Cold Start on Online Advertising Platforms: Data-Driven Algorithms and Field Experiments

Zikun Ye[1],Dennis J. Zhang[4], Heng Zhang[2], Renyu Zhang[3], Xin Chen[1], Zhiwei Xu
[1] University of Illinois at Urbana-Champaign, Urbana, IL
[2] Arizona State University, Tempe, AZ
[3] New York University Shanghai, Shanghai, China
[4] Washington University in St. Louis, St. Louis, MO
zikunye2@illinois.edu, denniszhang@wustl.edu, hengzhang24@gmail.com, renyu.zhang@nyu.edu, xinchen@illinois.edu

Cold start describes a commonly recognized challenge in online advertising platforms: With limited data, the machine learning system cannot accurately estimate the click-through rates (CTR) nor the conversion rates (CVR) of new ads and in turn cannot efficiently price these new ads or match them with platform users. Unsuccessful cold start of new ads will prompt advertisers to leave the platform and decrease the thickness of the ad marketplace. To address the cold start issue for online advertising platforms, we build a data-driven optimization model that captures the essential trade-off between short-term revenue and long-term market thickness of advertisement. Based on duality theory and bandit algorithms, we develop the Shadow Bidding with Learning (SBL) algorithm with a provable regret upper bound of $O\big(T^{\frac{2}{3}}K^{\frac{1}{3}}(\log T)^{\frac{1}{3}}d^{\frac{1}{2}}\big)$, where K is the number of ads and d is the effective dimension of the underlying machine learning oracle for predicting CTR and CVR. Furthermore, our proposed algorithm can be straightforwardly implemented in practice with minimal adjustments to a real online advertising system. To demonstrate the practicality of our cold start algorithm, we collaborate with a large-scale online video sharing platform to implement the algorithm online. In this context, the traditional single-sided experiment would result in substantially biased estimates. Therefore, we conduct a novel two-sided randomized field experiment and devise unbiased estimates to examine the effectiveness of the SBL algorithm. Our experimental results show that the proposed algorithm could substantially increase the cold start success rate by 61.62% while only compromising the short-term revenue by 0.717%, and consequently boost the total objective value by 0.147%. Our study bridges the gap between the bandit algorithm theory and the ads cold start practice, and highlights the significant value of well-designed cold start algorithms for online advertising platforms.

Key words: Cold Start Problem, Online Advertising, Contextual Bandit, Two-Sided Field Experiment

1. Introduction

With the rapid growth of internet technology and smartphone penetration, online advertising has become an unprecedentedly enormous industry with a substantial impact on the entire economy. The Interactive Advertising Bureau[1] reports that online advertising revenue of the United States has increased to $124.6 billion in 2019 (16% year-over-year growth rate compared to 2018, 19% average annual growth rate since 2010), 70% of which comes from mobile advertising. Indeed, the primary approach taken by large online platforms such as such as Facebook and TikTok to monetize

[1] See https://www.iab.com/insights/internet-advertising-revenue-fy2019-q12020/ for more details.

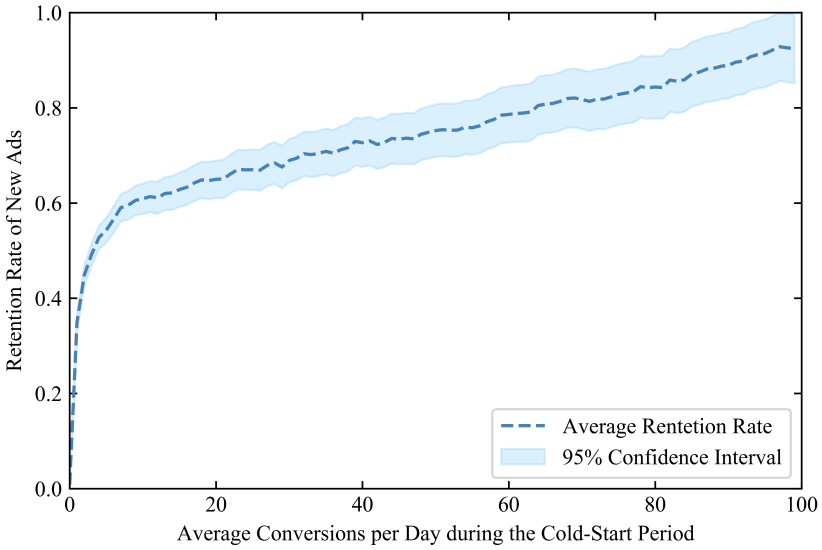


Figure 1 Retention Rate

their gigantic user traffic is online advertising. For example, in 2019, Facebook earns $69.7 billion revenue from advertising, which consists of 98.53% of its total revenue[2].

A core and well-known challenge faced by an online advertising system is the cold start problem (see, e.g., Dave and Varma 2014, Choi et al. 2020). Throughout the whole ad campaign, advertisers especially value the advertising performance in the first few days during which a bad performance with few conversions (e.g. app installs, purchases, etc) may result in advertisers' budget cut or even exodus. Therefore, it is crucial for a platform to help new ads gain the satisfying performance and to economically win the advertisers' loyalty and maintain the thickness of the ads pool in the auction (see Section 3 for more details). To illustrate that the cold start performance fundamentally impacts the long-term behavior of ads, we plot Figure 1 characterizing the relationship between the number of conversions during the first three days, referred to as the cold start period, of new ads (*i.e.,* the x-axis) and the retention rate of new ads in the next two weeks on the large-scale online video sharing platform we collaborate with (referred to as Platform O hereafter).[3] A key observation from Figure 1 is that the long-term retention rate and in turn the revenue of a new ad is positively correlated with its performance during the cold start period and such positive correlation is flattened when the number of conversions reaches a threshold around 10. This important observation indicates that for an ad to thrive and generate high revenues on the platform, quickly accumulating the first few conversions is essential. Not only is the ad retention dependent on the cold start, advertisers are also sensitive to whether their ads can obtain enough conversions during the cold start period.

[2] See the financial report of Facebook: https://www.sec.gov/ix?doc=/Archives/edgar/data/1326801/000132680120000013/fb-12312019x10k.htm.

[3] To protect the platform's identity and sensitive data, we re-scale the y-axis value to $[0, 1]$. The curve pattern remains the same if we vary the time length of "two weeks" from one day to thirteen days.

For Platform O, advertisers will carefully monitor the ad performance during the cold start period and may tighten the budget, reduce the ad materials, or leave the platform if they are unsatisfied with the performance. Therefore, the cold start performance would significantly impact both the thickness of the ad marketplace and the number of advertisers for an online platform.

On the other hand, the platform cannot simply provide more traffic to new ads during the cold start period since the platform has less information about these new ads during their cold start periods and in turn are less likely to efficiently match potential customers with these new ads. It has been widely noticed both in the literature and in practice that the limited data history would keep the platform from accurately predicting the click-through rate (CTR) and conversion rate (CVR) of the new ads (see, e.g., Choi et al. 2020). For Platform O, the area under the curve (AUC) of the new ads CTR prediction is significantly smaller than that of mature ads by 5.77% (to protect sensitive data, we only report the relative difference here), which is a sizable gap for a large-scale platform and indicates it is more difficult to predict the CTR of new ads than that of mature ones. This prediction inaccuracy is amplified by the sparsity of conversions. To address this issue, Facebook has suggested its advertisers to set enough budget for at least fifty conversions in order to successfully bring their ads out of the initial learning phase (i.e., the cold start period)[4]. The inaccuracy of CTR and CVR prediction for new ads naturally brings up the exploration-exploitation trade-off between the short-term revenue generated by mature ads (exploitation) and the long-term cold start value for new ads (exploration). *To sum up, the fundamental trade-off in solving the cold start problem is to dynamically balance the long-term gain of successfully cold starting new ads and the short-term dis-utility from inefficiently matching new ads during the cold-start period to optimize the long-run ad revenue.*

The main goal of this paper is to develop a new theoretically sound and practically feasible approach to solve the cold start problem for online advertising platforms. To this end, we build a novel data-driven optimization model that integrates both the short-term revenue and the long-term cold start reward of an online platform. With our model, we develop a primal-dual based multi-armed bandit (MAB) algorithm (denoted as the Shadow Bidding with Learning Algorithm), which adaptively adds a shadow bid to each new ad's bidding price. Our proposed algorithm well bridges theory and practice: It not only has a provable performance guarantee, but also could be straightforwardly implemented into the real-time bidding system of an online advertising platform in practice with minimal adjustment. To demonstrate the practical value of our algorithm, we collaborate with Platform O to conduct large-scale randomized field experiments to evaluate our algorithm. Our experimental results show that the proposed algorithm significantly increases the

[4] See https://www.facebook.com/business/help/112167992830700?id=561906377587030.

cold start value of new ads without compromising the short-term revenue much, so that the total objective value of the platform is boosted.

We summarize the main contributions of this paper as follows:

Modeling. To our best knowledge, we are the first in the literature to formulate the cold start problem for online advertising platforms as a data-driven optimization model, which synthesizes the linear program and MAB models in an innovative fashion. Previous research on cold start in advertising has mainly focused on improving the CTR, CVR prediction accuracy (Choi et al. 2020). The model, together with the associated algorithms, proposed in this paper, however, not only substantially improves the CTR/CVR prediction accuracy for new ads, but also provides a new lens to view and study the cold start problem for online advertising platforms as a (user impression) resource allocation (to new and mature ads) model under uncertainty. Therefore, our modeling framework has the potential to empower other studies of the cold start problem for online advertising and recommender systems from an optimization perspective.

Algorithm. We develop a novel Shadow Bidding with Learning (SBL) algorithm by embedding a linear program primal-dual framework into an ϵ-greedy contextual bandit algorithm. Though theoretically compelling, all existing algorithms for general contextual bandits with concave objectives (e.g., Agarwal et al. 2014, Agrawal et al. 2016) are practically infeasible on an online advertising platform. This is because these algorithms rely on an underlying *empirical risk minimization oracle*, which is unavailable or computationally inefficient in practice. Our proposed algorithm takes a different duality-based epsilon-greedy approach and bridges the gap between learning theory and ad cold start practice with a provable performance guarantee and straightforward implementation on real online advertising platforms. On one hand, the proposed algorithm leverages the dual variables of the cold start reward constraints, the power of the advertising platform's underlying machine learning models to predict CTR and CVR, and the adaptive exploration of the $\epsilon-$greedy algorithm, thus yielding a provable regret bound of $O\big(T^{\frac{2}{3}}K^{\frac{1}{3}}(\log T)^{\frac{1}{3}}d^{\frac{1}{2}}\big)$, where K is the number of ads and d is the effective dimension of the underlying machine learning oracle for predicting CTR and CVR. On the other hand, the SBL algorithm enables us to minimally adjust the real-time bidding system of an online advertising platform by simply adding the shadow bid of each ad (i.e., the dual variable of the cold start value constraint) to its system bidding price.

Experiment. To our best knowledge, we are the first in the literature to conduct two-sided randomized field experiments for bandit algorithms. In the ad cold start setting studied in this paper, the traditional one-sided experiment is invalidated by the violation of the Stable Unit Treatment Value Assumption (SUTVA) (see Section 5 for more discussions on this point) and, therefore, gives rise to estimation biases by as high as 120%. Such violation of SUTVA is common in the experimental evaluation of algorithms and policies on e-commerce (e.g., Facebook Marketplace,

see Ha-Thuc et al. 2020) and vacation rental (e.g., Airbnb, see Johari et al. 2020) platforms and has caused substantially biased estimations thereof. To address such a challenge, we design and implement novel two-sided field experiments on a large-scale video sharing platform (Platform O). The experiments restore SUTVA and enable us to causally estimate the value of our proposed algorithm in an unbiased fashion. The new experiment framework has the potential to be applied to evaluate other algorithms and policies of recommender systems on two-sided platforms. Based on our two-sided field experiments, we find that the proposed algorithm successfully increase cold start success rate by 61.62% and boost the total objective value by 0.15%. We emphasize that, given the gigantic scale of Platform O, such an increase would imply a sizable hundred-million-dollar boost in the ad revenue per year in the long-run. Furthermore, our experiments are conducted on only 33% of the user views and 20% of the ads. Because of the positive two-sided network effects between ads and users on Platform O, we would expect a much stronger enhancement of the total long-term value if our proposed SBL algorithm is applied to a larger user traffic and more ads.

To sum up, our study bridges the gap between the bandit algorithm theory and the practice of ad cold start, and highlights the significant value of well-designed cold start algorithms for online advertising platforms. The rest of this paper is organized as follows. In Section 2, we position our paper in the relevant literature. Section 3 discusses the business practices based on which we build our model. In Section 4, we propose our algorithm, analyze its regret bound, and report simulation-based numerical results. We introduce our field experiment setting in Section 5 and report our experimental results in Section 6. Section 7 concludes the paper by discussing promising directions for future research. All proofs are relegated to the Online Appendix.

2. Literature Review

Our paper is primarily related to three streams of literature: cold start for online advertising, bandit algorithms, and field experiments on large scale online platforms.

It has long been documented in the literature that estimating the CTR of new ads is a challenging problem in advertising because there is very little data and information to approach reliable prediction (see, e.g., Dave and Varma 2014, Choi et al. 2020). From the machine learning perspective, more sophisticated models with more features are designed to better estimate the new ads CTR. For example, Zhou et al. (2018) propose the deep interest network which incorporates historical behavior data of user interests to learn the CTR. Zhao et al. (2015) focus on cold-start product recommendation, using microblogging information, which increases AUC of new item CTR prediction significantly. Dave and Varma (2012) identify ad-hoc features of ads that could benefit CTR estimation from sparse and noisy data. In contrast, our proposed cold start algorithm (SBL) does not use any extra data or different neural network architectures. Instead, we employ the ϵ-greedy random exploration scheme and the shadow bids to feed more data of new ads into the

neural networks, which has also substantially increased the accuracy of CTR/CVR estimations. In addition, our work also provides a new perspective to study the cold start problem for online advertising platforms as a data-driven optimization model and offers efficient algorithms to tackle this challenge. The perspective to view the problem as user impression allocation is in line with another stream of literature that addresses ad allocation using mechanism design in the repeated auction setting (see, e.g., Caldentey and Vulcano 2007, Balseiro et al. 2015, Hojjat et al. 2017). In particular, Hojjat et al. (2017) investigate the ad allocation problem with guaranteed displays. Our algorithm can be readily extended into their setting by changing the target number of conversions into target number of impressions, and taking the parameter $\boldsymbol{\beta}$ to infinity (then, the target number of impressions for each ad will become a hard global constraint).

Our proposed algorithm builds upon ϵ-greedy contextual bandits and is related to four different bandit algorithms previously studied in the literature: (1) ϵ-greedy bandits. ϵ-greedy is one of the simplest exploration strategies and is equipped with an $O(T^{\frac{2}{3}})$ regret (Sutton and Barto 2018). One approach to deal with a complicated online learning problem is to reduce a bandit problem to supervised learning with simple exploration strategies, such as the epoch-greedy algorithm (Langford and Zhang 2007). Although this approach has a sub-optimal regret upper bound of $O((K\log|\Pi|)^{\frac{1}{3}}T^{\frac{2}{3}})$, where Π is the explored policy set, it makes minimum changes to a practical online advertising system. (2) Linear contextual bandits with upper confident bound (UCB) exploration. Linear bandits are successful in both theory and practice (Chu et al. 2011). However, in a real advertising system, the underlying models for predicting CTR and CVR are complicated neural networks, so the linear payoff assumption does not hold and LinearUCB can hardly be implemented. One remedy, called neural linear bandits, is to only explore the last linear layer of the neural networks (e.g., Riquelme et al. 2018). However, this method also makes significant changes to the online system and has no theoretical performance guarantee. (3) General contextual bandits. The recent advancement in general contextual bandits by Agarwal et al. (2014) covers a wide range of contextual bandits with optimal regret guarantee of $O(\sqrt{KT\log(T|\Pi|)})$, however, it requires solving optimization problems with a given *empirical risk minimization oracle* to balance the exploration and exploitation trade-off, with computation complexity $O((KT)^{\frac{3}{2}})$, the computation time of which is intensive in the online advertising system without the online oracle to solve the embedded optimization. This makes it infeasible to implement this algorithm and its follow-ups in a real online advertising platform. (4) Bandits with knapsacks. This variant of bandits is first proposed by Badanidiyuru et al. (2013). Recent works extend this algorithm to a general contextual setting with concave objectives (e.g., Agrawal et al. 2016). Our work is aligned with knapsack bandits on solving the primal-dual linear program to obtain the optimal policy. The regret of the bandit algorithms in this literature is benchmarked with the best policy in a

policy set, and dependent on the cardinality of the policy set. Furthermore, the expected reward of the algorithm is estimated with *Inverse Propensity Score*. Though theoretically convenient, such approaches make it difficult to implement the bandit algorithm on a practical large-scale advertising platform with the neural-network-based estimation of CTR/CVR. Following the idea of Foster et al. (2018), Foster and Rakhlin (2020), our SBL algorithm decomposes the problem into an ϵ-greedy bandit with the underlying prediction model, which could be quite general and takes the form of, e.g., linear regression, regression tree, and neural network. Our theoretical regret analysis with the underlying machine learning oracle being neural networks is based on the recent progress of *Neural Tangent Kernel* (Jacot et al. 2018, Arora et al. 2019). Our main algorithmic contribution towards the MAB literature is that we bridge the gap between theory and practice by developing the SBL algorithm with provable performance guarantee and straightforward implementation on real online advertising platforms.

Besides the general bandit algorithm literature, our work is also closely related to the growing literature on operations management problems with online learning. With the intrinsically uncertain and unknown environment in business, it is a recent trend to combine the learning theory and optimization to solve revenue management and pricing problems. For example, Chen et al. (2019) build an algorithm to solve the join pricing and inventory control problem with non-parametric demand learning for non-perishable products and show the regret convergence result. Nambiar et al. (2019) propose an algorithm with theoretical performance guarantees to solve the dynamic pricing problem with mis-specified demand models and evaluate its performance using offline simulations. Ferreira et al. (2018) use Thompson sampling to learn the demand at each price and solve the network revenue management problem. Chen et al. (2020) build an online learning algorithm to solve single-item inventory control problem under the periodic-review, backlogging policy with unknown capacity and demand distributions. Chen and Gallego (2018) propose a primal-dual learning algorithm to learn the dual optimal solution for personalized dynamic pricing problem with an inventory constraint. Bastani et al. (2019) propose a meta dynamic pricing algorithm to learn the prior through experiments while solving the pricing problem. Golrezaei et al. (2019) propose learning algorithms to set reserve prices in contextual auctions. Our main contribution towards this literature is that we have not only proved the theoretical performance guarantee of the proposed algorithm, but also implemented it on a large-scale advertising platform and tested its performance using field experiments.

Last but not least, our paper does directly relate to the growing literature on field experiments in online platforms (Terwiesch et al. 2019). For example, Zhang et al. (2020) document the spillover effects across platform users in a field experiment on a retailing platform. Zeng et al. (2020) show social nudge can boost the productivity of content providers on a social network platform through

randomized field experiments. Fisher et al. (2018) leverage both modeling and field experiments to study competition-based dynamic pricing in retailing. Several other papers in the literature also conduct field experiments to study platform operations problems (e.g., Cui et al. 2019a,b, Feldman et al. 2018). In the marketing literature, Schwartz et al. (2017) implement a learning algorithm to optimize the user acquisition strategy through display advertising and conduct a field experiment to gauge its effectiveness. Recently, there is a growing literature that examines the violation of SUTVA for experiments on large-scale two-sided platforms. Ha-Thuc et al. (2020) develop a new counterfactual framework for seller-side A/B testing on Facebook Marketplace and show that the new experiment framework satisfies the SUTVA. Johari et al. (2020) propose a mean-field model to show that single-sided experiment (demand-side or supply-side randomization) will result in biases in estimation for a two-sided marketplace like Airbnb. They also propose a two-sided randomization and the associated estimator which is unbiased when the supply and demand are extremely imbalanced. Our contribution towards this stream of research is the design and implementation of a novel two-sided randomized field experiment to causally estimate the value of our proposed bandit algorithm. The proposed experiment framework could potentially be applied to evaluating other algorithms and policies in general recommender systems of large-scale two-sided online platforms.

3. Background and Model

In this section, we first introduce the background setting of a typical Demand Side Platform (DSP) of online advertising, with particular focus on their auction mechanisms, billing options, and the Proportional–Integral–Derivative (PID) based bid control system. Based on the institutional knowledge of these settings, we then develop a data-driven optimization model that integrates linear program and multi-armed bandit to tackle the cold start issue on a DSP.

3.1. Online Advertising Platforms

A large-scale online platform such as Facebook or Twitter is usually equipped with a centralized advertising system called the DSP, which aggregates a lot of online ads and efficiently matches the ads with user views. In the model, we denote the online platform as "the platform" and its advertising platform as "the DSP". The overall landscape of a DSP can be summarized as Figure 2. Advertisers and platform users interact with each other on the DSP. On the demand side, advertisers set up their advertising campaigns by submitting the necessary information to the DSP, including bid prices, billing options, ad contents, and the advertising budget. On the supply side, platform users are exposed to ads while viewing the organic contents. With the huge traffic of the platform and the rich information of the advertisers and users, the DSP plays a central role to allocate user impressions to different ads with a goal of maximizing the long-term total revenue.

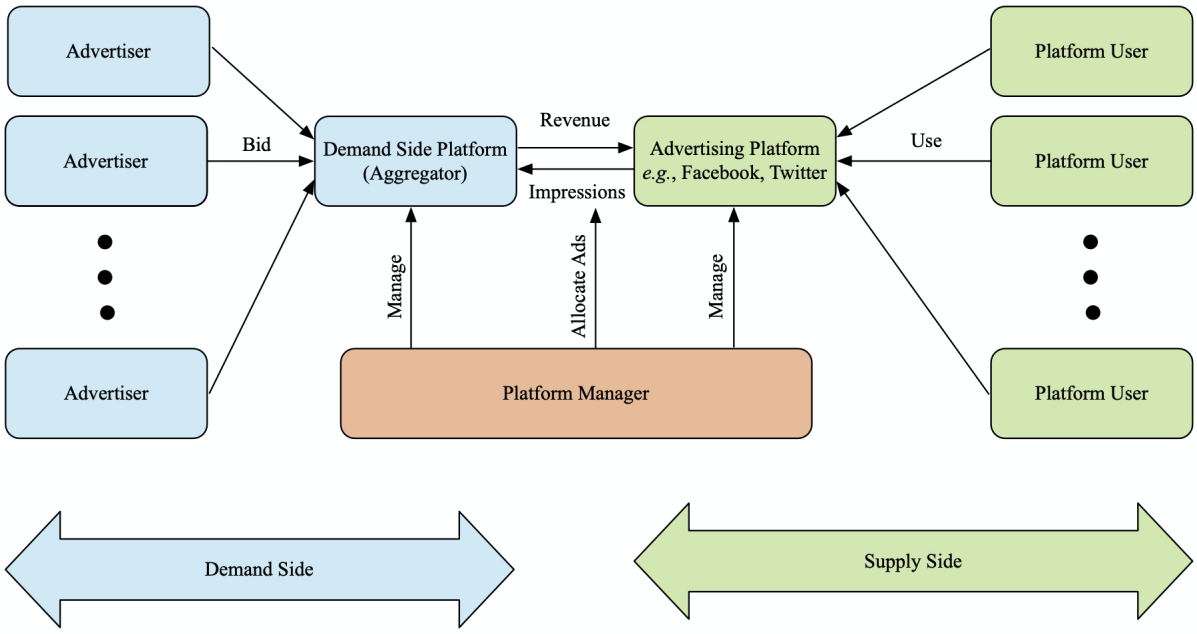


Figure 2 Illustration of the Online Advertising on Platforms

Next, we detail how the DSP monetizes, specifically, its auction mechanism, billing options, and PID-based bid control system.

Auction Mechanisms and Billing Options. Ad impression requests from platform users continuously arrive at the DSP. In the following paper, we use ad impression, user view and user impression interchangeably. As a consequence, for a large scale online platform, the DSP allocates billions of ad impressions to hundreds of thousands of ads each day. In order not to ruin the user experience, the DSP needs to efficiently match the tremendous number of ads and ad impressions within milliseconds. Before entering the auction stage, the DSP quickly downscales the size of the ad pool from hundreds of thousands to hundreds by simple filtering rules set by advertisers and predictive models. At the auction stage, hundreds of ads compete to win an ad impression based on advertisers' bids. The ad impression is allocated to the ad with the highest *estimated Cost per Mille* (eCPM) of the match, which measures the expected revenue of displaying the ad to the respective platform user for a thousand times. Such an allocation rule ensures that each ad impression generates the highest *ex ante* revenue in expectation.

The eCPM of a match between an ad and an ad impression depends on what the advertiser bids on (impression, click, or conversion). More specifically, if the advertiser bids on *impression*, eCPM is the bid itself (eCPM=bid). If the advertiser bids on *click*, eCPM equals the bid multiplied by the predicted CTR (pCTR) of the ad (eCPM=bid$\times$pCTR). Finally, if the advertiser bids on *conversion*, eCPM equals the bid multiplied by the product of pCTR and the predicted conversion

Table 1 Billing Options

Payment Scheme	Bid Price	Charged upon	Fee Deduction	eCPM (Rank by)
CPM	bid_impression	impression	bid_impression	bid_impression
CPC	bid_click	click	bid_click	bid_click×pCTR
oCPM	bid_conversion	impression	bid_conversion×pCTR×pCVR	bid_conversion×pCTR×pCVR
oCPC	bid_conversion	click	bid_conversion×pCVR	bid_conversion×pCTR×pCVR
CPA	bid_conversion	conversion	bid_conversion	bid_conversion×pCTR×pCVR

Note: This table is mainly for the first-price auction. Also, bid_impression, bid_click, bid_conversion are the bids on impressions, clicks, and conversions given by advertisers, respectively. The column Fee Deduction gives the budget depleted upon each impression, click, or conversion.

rate (pCVR) of the ad (eCPM=bid×pCTR×pCVR), where pCVR is defined as the rate of the user being converted after clicking the ad. Here, conversion means that, upon clicking an ad, a user eventually becomes the advertiser's customer. A typical conversion, sometimes also called an "action" of the user, may take different forms, such as app installation or deposit in a game.

Typically, there are several different billing options for advertisers to choose from, including *Cost per Mille* (CPM), *Cost per Click* (CPC), *Cost per Action* (CPA), *Optimized Cost per Mille* (oCPM) and *Optimized Cost per Click* (oCPC), the differences among which we summarize in Table 1. Under the CPM, CPC, and CPA billing option, advertisers bid on the impressions, clicks, and conversions, respectively, and are directly charged after their ads are displayed, clicked, or converted. Due to the intrinsic uncertainty of user clicks and conversions, the advertiser bears a high risk under the CPM scheme. On the other hand, if an ad is displayed to a platform user, it already makes a negative impact on user experience and causes losses to the platform. As a consequence, in the current online advertising market (such as Facebook and Platform O), oCPM and oCPC under which the DSP and the advertiser *share* the click and conversion uncertainty risks are the most popular billing options. More specifically, under both oCPM and oCPC, the advertisers bid on conversion. However, the advertiser will be charged by the expected cost per impression, bid_conversion×pCTR×pCVR (resp. the expected cost per click, bid_conversion×pCVR), when the ad is displayed to (resp. clicked by) a user under oCPM (resp. oCPC). One should note that, because of the randomness in click-through and conversion rates, the actual payment of the advertiser per conversion is not necessarily the same as its bid for a conversion under oCPM or oCPC. For each billing option, the auction may run in a first-price or second-price fashion, under which the winning advertiser pays its own bid, or the bid with the second-highest eCPM.

PID-based Bidding System. As discussed above, both oCPM and oCPC suffer from the issue that the actual cost (per conversion) of an advertiser is different from its bid (also known as the target cost). Such a cost-control issue is exacerbated by the following: (a) second-price auction, under which the winning advertiser pays the expected cost per impression/click of the bidder with the second-highest eCPM; and (b) biased estimation of pCTR and pCVR, under which for each ad and ad impression pair the DSP could not accurately estimate the CTR and CVR. In practice,

the PID controller is widely adopted on online advertising platforms (Yang et al. 2019, Zhang et al. 2016) to control the gap between the actual cost and the target cost for an advertiser. Under PID, advertisers authorize the DSP to adaptively change their real-time bid prices to address the aforementioned *cost control problem*. The core of the PID controller is a simple feedback control idea: If the actual cost per conversion falls below the target cost, the DSP will increase the bid price thus boosting its eCPM and the chance of winning the auction, and vice versa. This real-time bid price given by the PID controller is referred to as the *System Bidding Price*. We leave more details about the PID system to Appendix C.

Cold Start on a DSP. As discussed in the Introduction, the cold start of new ads is one of the key challenges faced by both platforms and advertisers. It is extremely difficult to strike a smart balance between boosting new ads with high potentials to enhance the long-term thickness of the platform and maximizing the short-term revenue generated by mature ads with high quality. To the best of our knowledge, most DSPs tackle the ad cold start problem in an ad hoc fashion. For example, to increase the cold start success rate, the current practice of Platform O is adopting the PID controller to *uniformly* increase the system bidding prices for all new ads within a very short time horizon until the upper bounds of the system bidding prices are met. This approach increases the probability of winning impressions for new ads, which results in more explorations of the new ads and potentially more conversions. Soon after such sharp increases in the system bidding prices, the PID system will adaptively lower the system bidding prices to offset the extra costs caused by the uniform bid increases. To our knowledge, this heuristic approach has no performance guarantee, with fine-tuning of hyper-parameters as the only leverage.

3.2. Cold Start Problem Setting

We formulate the cold start problem of a DSP as a data-driven optimization model. To highlight the key trade-off associated with the cold start problem and avoid unnecessary complexity, we make two high-level modeling assumptions. First, the ad allocation mechanism is a first-price auction with the oCPC billing option, which is the current practice of the DSP we work with. Without loss of generality and for the ease of exposition, we assume CVR=pCVR= 1, i.e., conversion is guaranteed upon a customer's click-through. As will be clear in our online implementation, our proposed algorithm can be easily implemented for the real setting where CVR and pCVR is small. The first-price auction is more intuitive for advertisers, so there is a recent trend of switching from second-price auctions to first-price auctions in the online advertising industry. For example, Google Ad Manager has moved to the first-price auction in 2019[5]. Second, the system bidding price of each ad remains the same as the bid submitted by the advertiser. In other words, for our theoretical

[5] https://www.blog.google/products/admanager/rolling-out-first-price-auctions-google-ad-manager-partners/

model, we set $k_p = k_i = k_d = 0$ in the PID system (see Appendix C). In the online implementation of our proposed algorithm, the model is adapted to incorporate the actual online auction mechanism and the real-time system bidding prices of the DSP we experiment on.

For ease of reading, we list all the notations of the paper in Appendix A, Table 5. We consider a DSP where a set of K new ads, denoted as $A := \{1, 2, ..., K\}$, are competing for user impressions. We only consider new ads in our base model and will discuss how our algorithm can be generalized to a setting with both new and mature ads in Section 4.3. User views arrive at the DSP in a sequential manner and we define the set of all user views as $[T] := \{1, 2, ..., T\}$. For each user view t, there is an associated context vector $x_t \in X$, where X is a countable feature space. The context x_t could be quite general, containing the demographic and behavioral information of user view t inherited from the platform. Upon the arrival of user view t, the DSP observes the context x_t. Suppose that ad $a_t \in A$ is chosen to be displayed. We can define a $K-$dimensional binary vector $\boldsymbol{v}_t(a_t) \in \{0,1\}^K$ representing whether each ad is clicked. More specifically, $v_{tj}(a_t) = 1$ only if $a_t = j$ and ad j is clicked by user t. Furthermore, we assume a *stochastic bandit setting*, i.e., for any $a_t \in A$, $(x_t, \boldsymbol{v}_t(a_t))$ is drawn *i.i.d.* from a distribution $\mathcal{D}$ over $X \times \{0,1\}^K$, which is unknown to the DSP. We denote its marginal distribution over X as $\mathcal{D}_X$. Given $x_t = i$ and $a_t = j$, we define $c_{ij} := \mathbb{E}[v_{tj}(a_t)|x_t = i]$ as the CTR of ad j for user view t. We sometimes also abuse the notation and denote by c_{tj} the CTR of ad j for round t, given the context information for round t.

A core challenge faced by Platform O and other online advertising platforms is jointly optimizing the revenue and the cold start value of new ads. To evaluate the revenue during the cold start period, we define $\boldsymbol{V} := \sum_{t=1}^{T} \boldsymbol{v}_t(a_t)$ as the accumulative click-through vector, where $V_j := \sum_{t=1}^{T} v_{tj} \in \{0, 1, ..., T\}$ is the total number of clicks generated by ad j until customer T. As prescribed by the oCPC billing option, the total revenue generated by the ads is given by

$$\sum_{j=1}^{K} b_j V_j,$$

where $b_j \in [0, 1]$ is the bid (per click) of ad $j \in A$. To quantify the cold start value, one may want to directly estimate the total life-time revenue from an ad based on the number of accumulated conversions during its cold start period (the first 3 days for Platform O). However, such an estimation is extremely difficult, if not impossible, because we need to establish the causal effect of conversions during the cold start period on the life-time revenue of the new ad. Therefore, we take an alternative approach to approximate the aforementioned relationship between conversions in the cold start period and the life-time revenue. We observe from Figure 1 that the retention of an ad increases linearly in the number of clicks/conversions before this number is below a certain threshold and stays (almost) unchanged once it exceeds the threshold. Motivated by this phenomenon,

we assume the cold start value of each conversion for ad j before the number of accumulated conversions reaching the conversion target as $\beta_j \in (0,1]$. Without loss of generality, we denote the conversion target as αT, where $\alpha \in (0,1)$. Thus, the cold start reward is given by

$$\sum_{j=1}^{K} \beta_j \min\{V_j, \alpha T\}.$$

In practice, the conversion target αT is determined by business practice and validated by our observation in Figure 1. We specify the cold start value per conversion β_j via two steps: (a) Inherit the business practice of Platform O that $\beta_j = 2b_j$ for each ad j, and (b) conduct simulations to validate the choice of $\boldsymbol{\beta}$. Our simulation results, Figure 8 in Appendix D, demonstrate that setting $\beta_j = 2b_j$ for each ad j would indeed result in the highest expected long-term revenue for the platform.

We are now ready to present the objective of the DSP, which equals the sum of revenue and cold start reward:

$$\Gamma(\boldsymbol{V}) := \sum_{j=1}^{K} b_j V_j + \sum_{j=1}^{K} \beta_j \min\{V_j, \alpha T\} = \sum_{j=1}^{K} b_j \sum_{t=1}^{T} v_{tj} + \sum_{j=1}^{K} \beta_j \min\left\{\sum_{t=1}^{T} v_{tj}, \alpha T\right\} \tag{1}$$

Notice that the cold start reward term of ad j is piece-wise linear in its number of conversions. Based on Figure 1 and the current practice of Platform O, we may also use the (weighted) cold start success rate as the cold start reward. Formally, the cold start reward can be evaluated as $\alpha T \sum_{j=1}^{K} \beta_j \mathbb{I}_{\{V_j \geq \alpha T\}}$ and the associated objective is given by $\Gamma_0(\boldsymbol{V}) := \sum_{j=1}^{K} b_j V_j + \alpha T \sum_{j=1}^{K} \beta_j \mathbb{I}_{\{V_j \geq \alpha T\}}$. Finding the policy π that maximizes $\mathbb{E}_{\mathcal{D},\pi}\big[\Gamma_0(\mathbf{V})\big]$ is generally NP-hard. Meanwhile, $\Gamma(\cdot)$ should approximates $\Gamma_0(\cdot)$ well in practice, and a common approach in high-dimensional statistics and machine learning literature is to use L_1 norm for an approximation of the L_0 norm (Hastie et al. 2015). In our field experiments (Section 6), we will report the value of our proposed algorithm with respect to both $\Gamma(\cdot)$ and $\Gamma_0(\cdot)$ for completeness.

3.3. Regret Definition

In this subsection, we formally define the benchmark for our proposed bandit algorithms. In every round $t \in [T]$, a policy π observes the context $x_t \in X$ associated with user t, chooses an ad/action $a_t \in A$ (possibly at random), and observes the random outcome whether the ad is clicked. We define the history update to round t as $\mathcal{H}_t = \bigcup_{s=1,\ldots,t-1}\{(x_s, a_s, \boldsymbol{v}_s(a_s))\}$. Let $\Delta_A := \{\boldsymbol{y} \in \mathbb{R}^{|A|} : y_j \geq 0, \forall j \in A, \sum_{j\in A} y_j \leq 1\}$ be the set of the non-negative weight/distribution over arms. Formally, a policy π defines a mapping from the history $\mathcal{H}_t$ and the context x_t to the set of distribution over arms Δ_A for any t.

Recall that one can express the expected reward we gain from policy π as $\mathbb{E}_{\mathcal{D}^T,\pi}\big[\Gamma(\boldsymbol{V})\big]$. We notice that $\Gamma(\cdot)$ is concave in $\boldsymbol{V}$. One can show using Jensen's Inequality that the following lemma holds, whose proof is relegated to Appendix B.1.

LEMMA 1. *For any policy π, the scaled expected reward can be upper-bounded as*

$$\frac{1}{T}\cdot\mathbb{E}_{\mathcal{D}^T,\pi}\big[\Gamma(\boldsymbol{V})\big]\le \mathsf{OPT}:=\max_{\boldsymbol{y}_i\in\Delta_A}\left\{\sum_{j=1}^{K}\mathbb{E}_{i\sim\mathcal{D}_X}\left[c_{ij}y_{ij}b_j\right]+\sum_{j=1}^{K}\beta_j\min\left\{\mathbb{E}_{i\sim\mathcal{D}_X}[c_{ij}y_{ij}],\alpha\right\}\right\}.$$

Essentially, OPT is the upper bound of the cold start objective function $\Gamma(\cdot)$, and can be viewed as the solution to a fluid version of our cold start problem, in which the decision variable y_{ij} is a sampling distribution over the ads j given a user context i. Similar upper bounds are widely used in the revenue management literature (Gallego and Van Ryzin 1994, Golrezaei et al. 2014, Zhang et al. 2018) as well as the bandit learning literature (Badanidiyuru et al. 2013, Agrawal et al. 2016). With Lemma 1, one can formally define the regret for an arbitrary policy π as

$$\mathsf{Reg}(\pi)=T\cdot\mathsf{OPT}-\mathbb{E}_{\mathcal{D},\pi}\big[\Gamma(\boldsymbol{V})\big]. \tag{2}$$

Our goal is to propose a novel policy with both provable performance guarantee (measured by a sublinear regret) and effective implementation on a practical DSP. Although the regret defined by (2) is common in the stochastic bandit setting with a concave objective function, (see, e.g., Agarwal et al. 2014, Agrawal et al. 2016), the existing bandit algorithms with concave objective functions are not practically feasible for our problem due to the following reasons. First of all, these algorithms are often built upon an offline cost-sensitive classification oracle for the policy class, which is intractable for most policy classes. In practice, good policy classes are usually hard, if not impossible, to construct. Furthermore, a practical DSP often relies on machine learning algorithms to generalize the knowledge learned from the observed click-through and conversion data, and make accurate predictions about future user behaviors. Second, existing MAB algorithms usually provide an empirical estimate of OPT, and then randomize over different policies (Agrawal et al. 2016) to approach the upper bound. In principle, such randomization techniques may also be extended to our model as well. For example, one may make use of the prediction power of machine learning and design algorithms that randomize over the new ads. However, such randomization scheme involves very substantial memory (scales linearly with T) and engineering efforts to completely revamp the existing DSP system (essentially a large-scale marketplace for ad auctions), which is literally impossible in practice. To sum up, existing bandit algorithms for concave objective functions are not compatible with the DSPs in practice. As such, we leverage the machine learning system of a DSP and design a novel primal-dual based algorithm that adds "shadow bids" to new ads, which can be easily integrated into the existing bidding system of a DSP. As demonstrated in Section 5, one can easily implement this "shadow bidding with learning" algorithm on a real-world DSP and our field experiments show significant improvements in the total objective function and long-term cold start value without much compromising the short-term revenue.

4. Cold Start Algorithm

In this section, we propose a novel bandit learning algorithm for our ad allocation model with cold start reward, which leverages the ϵ-greedy exploration strategy, the prediction power of a DSP's underlying machine learning models, and the empirically optimal dual solution to the fluid upper bound. In addition to proving the sublinear regret bound of our proposed algorithm, we also discuss how to implement the algorithm in a practical setting with a mixture of new and mature ads.

4.1. Shadow Biding with Learning (SBL) Algorithm

In this part, we outline our primal-dual based learning algorithm. In designing the algorithm, one central difficulty is the unknown distributional information of the underlying model. In particular, the CTRs at round t are unknown to any online algorithm. Instead, we only have access to an empirically estimated CTR via the online training of predicting models on the historical data. To get an empirically optimal ad allocation policy, one can solve the following ad allocation model at round t:

$$\max_{y_{sj}\geq 0,\ \sum_{j\in A} y_{sj}\leq 1} \sum_{s\leq t-1}\sum_{j\in A} \hat{c}_{ij}^s b_j y_{sj} + \sum_{j\in A}\beta_j \min\left\{\sum_{s\leq t-1}\hat{c}_{ij}^s y_{sj}, \alpha(t-1)\right\}. \tag{3}$$

Here, the empirical CTR $\hat{c}_{ij}^s$ is the estimated CTR at time s with the realized context i, and ad j trained on the history data $\mathcal{H}_{s-1}$. Comparing (3) to (1) reveals that, for the empirical ad allocation problem, we re-scale the time-horizon from T to $t-1$ and use the empirical estimation $\hat{c}_{ij}^s$ in place of the true CTR c_{sj}. By introducing an additional variable u_j for each ad j, we can transform (3) to a linear program:

$$\begin{aligned} \max_{y_{sj}\geq 0,\ u_j\geq 0} \quad & \sum_{s\leq t-1}\sum_{j\in A} \hat{c}_{ij}^s b_j y_{sj} + \sum_{j\in A}\beta_j[\alpha(t-1)-u_j] \\ \text{s.t.} \quad & \sum_{j\in A} y_{sj}\leq 1,\ \forall s\leq t-1,\ \sum_{s\leq t-1}\hat{c}_{ij}^s y_{sj}+u_j\geq \alpha(t-1),\ \forall j\in A \end{aligned} \tag{4}$$

We succinctly write down the dual of (4) as

$$\min_{\lambda_j\in[0,\beta_j]} \quad \sum_{s\leq t-1}\max_{j\in A}\left\{\hat{c}_{ij}^s(b_j+\lambda_j)\right\} - \alpha(t-1)\sum_{j\in A}\lambda_j, \tag{5}$$

which is a non-smooth convex program with decision variables $\boldsymbol{\lambda}=(\lambda_1,\lambda_2,\cdots,\lambda_K)$. Strong duality dictates that the optimal values of (4) and (5) must be the same. Utilizing such duality, we propose the following cold start algorithm.

Shadow Bidding with Learning (SBL)

Parameters: Epoch schedule $0=\tau_1<\tau_2<\ldots$ such that $\tau_m-\tau_{m-1}=\tau_{m+1}-\tau_m\leq O(T^{\frac{2}{3}})$. Cold start value coefficient $\boldsymbol{\beta}$. Target conversion parameter α.

Initialization: $\boldsymbol{\lambda}^0=0$, $t=0$, $m=1$

For $t=1,2,\ldots,T$ **do**

Step 1: Observes the context i at period t. With probability $\epsilon_t = t^{-\frac{1}{3}}(K\log t)^{\frac{1}{3}}$, the algorithm picks an ad uniformly at random. Otherwise, display an ad $a_t \in \arg\max_j \hat{c}_{ij}^t(b_j + \lambda_j^{\tau_m})$ with arbitrary tie-breaking rules.

Step 2: If $t = \tau_m$, we solve the following dual model to update $m := m+1$ and $\boldsymbol{\lambda}^{\tau_m}$ by the sub-gradient descent algorithm.

$$\min_{\lambda_j \in [0,\beta_j]} \quad \sum_{s\le t-1} \max_{j\in A}\{\hat{c}_{ij}^s(b_j+\lambda_j)\} - \alpha(t-1)\sum_{j\in A}\lambda_j, \tag{6}$$

Step 3: Observe the outcome of a_t, and update the parameters of the underlying machine learning model for predicting $\hat{c}_{i,j}^{t+1}$.

Several remarks are in order. First of all, in this algorithm, we periodically solve optimization problem (6) to produce the dual vector $\boldsymbol{\lambda}^{\tau_m}$. With the most recent $\boldsymbol{\lambda}^{\tau_m}$, and the most up-to-date CTR estimation given the context, $\hat{c}_{tj}$, the SBL algorithm picks ad j with the highest $\hat{c}_{tj}(b_j+\lambda_j^{\tau_m})$, with ties broken arbitrarily. Solving the dual problem with a carefully chosen epoch will save the computing resources without hurting the algorithm performance.

Second, the term *shadow biding* comes from the ad selection rule. We pick the ad with the largest "long-term" value upon the arrival of each user, which is sum of the estimated expected short-term revenue, $\hat{c}_{tj}b_j$, and the shadow price for the cold start reward, $\lambda_j^{\tau_m}$, multiplied by the predicted CTR. The original bidding process of the DSP only seeks to maximize the short-term revenue by picking ad j with the highest $\hat{c}_{tj}b_j$, whereas we add $\lambda_j^{\tau_m}$, the shadow price associated with the constraint $\sum_{s\le t-1}\hat{c}_{ij}^s y_{tj} + u_j \ge \alpha(t-1)$ in (4), to the bid b_j in order to capture the long-term cold start value of displaying ad j immediately. Alternatively, one may view (4) as an ad assignment problem, similar to the ones studied in Devanur and Hayes (2009) and Agrawal et al. (2014). Complimentary slackness implies that if $y_{\ell j} > 0$, then $\ell \in \arg\max_j \hat{c}_{tj}(b_j+\lambda_j^{\tau_m})$. In effect, we use the solution in the dual space to characterize the correct assignment in the primal space, which gives a fast and simple allocation rule. Similar dual-based strategies are used in the online linear program under stochastic input or random permutation (Li et al. 2020), and the non-contextual knapsack bandit setting (Badanidiyuru et al. 2013). One may also wonder whether existing algorithms that directly solve the primal problem (e.g., Agarwal et al. 2014, Agrawal et al. 2016) would also work in our ad cold start setting. In fact, though theoretically possible, directly implementing the primal solutions on a practical DSP is very hard, if not impossible. More specifically, solving the primal problem amounts to dictating an ad assignment scheme. However, the primal space of the problem is extremely large, whose solution has a cardinality of the number of impressions multiplied by the number of ads, which is at the order of trillions. The cardinality of the associated dual space, however, is the number of ads, which is at the order of hundreds of thousands. Therefore, working with the dual space is substantially simpler than working with the primal space. Furthermore, we will show in Section 5 that our dual-based solution (the SBL algorithm) can be easily adapted into

a practical DSP with a mixture of various bidding mechanisms (first-price, second-price, with or without the PID control system, etc.).

Third, in each round t, we explore new ads with probability $t^{-\frac{1}{3}}(K\log t)^{\frac{1}{3}}$, and exploit, with probability $1-t^{-\frac{1}{3}}(K\log t)^{\frac{1}{3}}$, following the dual-based policy (6). This exploration-exploitation schedule is common for ϵ-greedy algorithms in the bandit learning literature. The novelty of our algorithm lies in the well-designed dual-based exploitation scheme and the integration of an MAB algorithm with the underlying machine learning oracle of a DSP. One may also consider other exploration-exploitation strategies such as *Upper Confidence Bound* or *Thompson Sampling*. However, an ϵ-greedy based algorithm (such as SBL) can be naturally embedded into a current DSP without much engineering change, as we will show in Section 5.

4.2. Regret Bound Analysis

Following Langford and Zhang (2007), the majority of recent contextual bandit algorithms are based on a specific underlying optimization model, whose offline solution oracle is given. The recent progress on contextual bandits develops both computationally efficient and optimal learning algorithms based on the empirical risk minimization oracle (Agarwal et al. 2014, Agrawal et al. 2016) which leads to a regret of $\tilde{O}(\sqrt{KT\log(|\Pi|)})$. Specifically, under a fixed policy set Π and the set $\mathcal{S}$ of context-loss pairs $(x,\ell)\in X\times\mathbb{R}^K$, the oracle returns the loss-minimization policy $\pi^*=\arg\min_{\pi\in\Pi}\sum_{(x,\ell)\in\mathcal{S}}\ell(\pi(x))$. Our problem setting is fundamentally different from those in the literature in two aspects. First, we explicitly construct a very general (and very large) policy set Π for developing our algorithm. Commonly used policy sets in practice such as linear predictors, regression trees, and neural networks are typically with an extremely large cardinality. For example, if the policy set Π is the collection of neural networks with fixed structure, depth, and width, even under the proper parameter discretization, the cardinality $|\Pi|$ grows exponentially with the number of parameters. Second, in our setting, a policy π contains two sequential decisions: the CTR prediction and ad allocation decisions, which brings challenges in both computation and analysis. This two-step procedure combining data-driven estimation and optimization model is widely used in practice-based research in the operations literature (e.g., Glaeser et al. 2019, He et al. 2020, Bimpikis et al. 2020). In this regard, the learning algorithms established in the literature (e.g., Agarwal et al. 2014, Agrawal et al. 2016, with a regret depending on $\log(|\Pi|)$ and benchmarked with the best policy in set Π) do not apply in our setting. Instead, the optimal policy we benchmark with is the (relaxed) optimal primal allocation policy (the primal integer decisions $\{0,1\}^K$ relaxed to $[0,1]^K$) with the optimal CTR prediction model.

Unlike the empirical regret of a policy computed via *Inverse Propensity Score* (IPS) in most of the existing contextual bandit literature (Agarwal et al. 2014, Agrawal et al. 2016), we adopt the *Direct Method* (DM), which uses empirical estimates from the CTR prediction model to evaluate

the ad allocation policy. The IPS method gives an unbiased reward estimator of a policy and is, thus, widely used in regret analysis. However, IPS suffers from a high variance when the policy set is large and/or the past sample paths vary significantly, which is indeed the case of our real implementation on Platform O. However, the DM can be robustly implemented on the large-scale DSP and does not affect the current CTR prediction system thereof. As one may expect, under the DM, the total regret depends on the accuracy of the underlying prediction model. In our regret analysis, we demonstrate that as long as the ground-truth CTR can be well captured by the prediction model with a high probability, the SBL algorithm is asymptotically optimal. For more theoretical and computational comparisons between DM and ISP, we refer interested readers to, e.g., Dudík et al. (2011), Foster et al. (2018), Foster and Rakhlin (2020).

Notice that in running the algorithm, we are effectively solving

$$\mathsf{OPT}^t = \min_{0\le\lambda_j\le\beta_j} \sum_{i\in X} \hat{p}_i^t \max_{j=1,2,\dots,K} \left(\hat{c}_{ij}^t(b_j+\lambda_j)\right) + \alpha \sum_{j=1}^{K} (\beta_j - \lambda_j), \tag{7}$$

where $\hat{c}_{ij}^t$ is the estimate of c_{ij} produced by the underlying prediction model prior to round t. $\hat{p}_i^t$ denotes the empirical distribution of contexts prior at round t. Before formally presenting our main regret analysis result, we address two basic issues regrading this formulation. First, we need to bound the gap between optimal empirical *primal* allocation and our optimal empirical *dual* allocation. By strong duality, this gap is induced by tie breaking in Steps 1 and 2 of the SBL algorithm. As we will show in Appendix B, adding an arbitrarily small perturbation to the CTR estimate $\hat{c}_{ij}^t$ will ensure that the tie-breaking in Step 1 will only induce an arbitrarily small additional regret. To bound the gap from tie breaking in Step 2, we make the following assumption.

Assumption 1. *For each context $i \in X$ and each ad $j \in A$, it holds that*

$$\hat{p}_i^t \hat{c}_{ij} \le O(T^{-\frac{1}{3}} (\log T)^{\frac{1}{3}} K^{-\frac{5}{3}}).$$

Assumption 1 states that the empirically estimated probability of a user with context i clicking ad j is negligible. This assumption is introduced to guarantee that the error from tie breaking in Step 2 of the SBL algorithm is small. Similar assumptions are made for other primal-dual settings (e.g., Devanur and Hayes 2009, Agrawal et al. 2014). We note that a typical online DSP faces hundreds of millions of different users and each of them can be regarded as a unique context. Therefore, Assumption 1 is made without loss of generality in practice. We remark that the sub-gradient descent approach in Step 2 of SBL may also incur some computational errors. To demonstrate the effectiveness of the sub-gradient descent algorithm in our ad allocation model, in Appendix E, we report a numerical experiment which shows the solution of our primal-dual subgradient descent

process produces almost the same objective function value as the exact solution in the primal space by the Simplex method.

The second issue of the SBL algorithm is the prediction error associated with the underlying machine learning model to estimate CTR. Clearly, the performance of our algorithm depends on that of the underlying predictor for estimating CTR. In practice, the underlying predictor returns the predicted CTR $\hat{c}_{ij}$ by training from a class of functions $\mathcal{X}$ which estimates the CTR of each context facing each ad ($X \times A \mapsto [0,1]$). The CTR predictor may take the form of linear regressors, regression trees, and neural networks, the last of which are the case with Platform O. To bound the prediction error of the underlying machine learning model, we make the following *Prediction Oracle* assumption.

ASSUMPTION 2 **(Prediction Oracle)**. *For each ad j with n_j^t observed i.i.d. contexts drawn from the context distribution $\mathcal{D}_{\mathcal{X}}$ before round t and the click-through outcomes of showing ad j to these contexts, with probability at least $1-\delta$, the estimate $\hat{c}_{ij}$ satisfies $|\hat{c}_{ij}^t - c_{ij}| \leq O\left(\sqrt{\log(1/\delta)d/n_j^t}\right)$ for any context i and $\delta \in (0,1)$, where d is the effective dimension of the predictor.*

The prediction oracle assumption is made without loss of generality and can be satisfied by most of the commonly used predictors in the literature and in practice, such as linear regressors, regression trees, and neural networks. We first note that, in the non-contextual setting (i.e., X is a singleton), Assumption 2 is reduced to the standard Hoeffding's inequality with the effective dimension $d=1$. If $\mathcal{X}$ is the set of linear regressors and the true data generating process is indeed a linear function, the prediction oracle assumption holds with the ridge regression and the effective dimension d defined as the context dimension (Chu et al. 2011). For the regression tree predictor, it has been well established in the literature (e.g., Wager and Walther 2015) that an adaptive regression tree with each child node containing at least η fraction of the data points in its parent node and each leaf node containing q training samples, has a more general convergence rate of $\sqrt{\frac{\log(n_j^t)\log(d)}{q\log((1-\eta)^{-1})}}$. Therefore, Assumption 2 is satisfied under the mild condition that the regression trees have a fixed depth and $q=\Omega(n_j^t)$, which commonly holds in practice. For a large-scale DSP in practice like Platform O, $\mathcal{X}$ is the set of fully connected neural networks with the ReLU as the activation function. We denote L as the network depth, w as the number of hidden nodes the same in each layer (i.e. the network width), and w_0 as the dimension of the feature space for contexts, i.e., $x_i \in \mathbb{R}^{w_0}$ for all $i \in X$. Following the convention of the neural network literature (e.g., Chizat et al. 2019), we parameterize the neural network by $\theta \in \mathbb{R}^d$, where the effective dimension $d = O(w^2L + ww_0)$. In Appendix G, we show that under the mild technical Assumption 3, with high probability there exist neural networks with such a parameterization and large enough width

w, satisfying the prediction oracle in both *lazy training* (Proposition 1) and the gradient based training (Proposition 2).

We are now ready to state our main theoretical result as follows.

THEOREM 1 **($\tilde{O}(T^{\frac{2}{3}})$ Regret Bound).** *Under Assumptions 1 - 2, the expected regret of the Shadow Bidding with Learning algorithm is bounded by* $O(T^{\frac{2}{3}}K^{\frac{1}{3}}(\log T)^{\frac{1}{3}}d^{\frac{1}{2}})$.

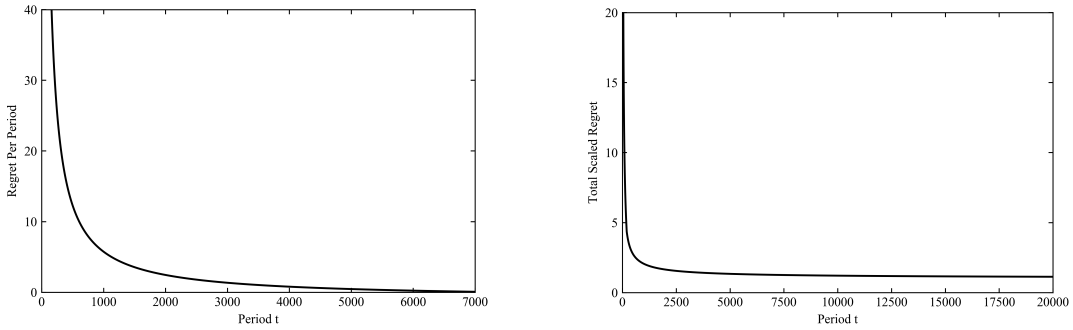
Theorem 1 shows that our proposed SBL algorithm has a regret of order $\tilde{O}(T^{\frac{2}{3}})$, which is consistent with the ϵ-greedy type algorithms for contextual bandits. Furthermore, the bound depends on the effective dimension of the predictor by $\sqrt{d}$. This is a natural and necessary price we have to pay with the SBL algorithm, which relies on the underlying machine learning model to predict CTR. If the underlying CTR prediction is easy (resp. hard) so that the effective dimension of the predictor is small (resp. large), our algorithm can achieve a sharper (resp. looser) regret bound. Theorem 1 presents our regret bound in the expected regret, but we can easily extend it to a high-probability type bound using the Azuma-Hoeffding Inequality. We also remark that the analysis of our ϵ-greedy based SBL algorithm is substantially more difficult than the standard contextual bandit algorithms with a linear reward function (e.g., Chu et al. 2011). This is because, in our problem, the cold start reward collected in the current round depends on the history. Furthermore, the dual-based bidding strategy, albeit easy to implement on a real DSP, is hard to analyze.

To prove Theorem 1, one has to carefully map the total reward collected so far into the dual space. More specifically, we first establish, in Lemma 2 (in Appendix B), the approximate complementary slackness and bound the duality gap between the empirical primal and the empirical dual due to tie breaking in Step 2 of the SBL Algorithm by $O(T^{-\frac{1}{3}}(\log T)^{\frac{1}{3}}K^{\frac{1}{3}})$. Then, we build an auxiliary reward process independent of history: Each click of ad j generates a reward of $b_j+\beta_j$, irrespective of whether the threshold αT is met. Based on the approximate complementary slackness and Hoeffding's inequality, Lemma 3 (in Appendix B) bounds, under the SBL algorithm, the gap between the auxiliary reward process and the optimal reward by $O(T^{\frac{2}{3}}(\log T)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}})$. Finally, in Lemma 4 (in Appendix B), we bound the gap between the auxiliary reward process and the true reward process by $O(\sqrt{KT\log T})$. Putting these bounds together would prove the desired regret bound of $O(T^{\frac{2}{3}}(\log T)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}})$ for the SBL algorithm. We refer interested readers to Appendix B for the proof details of Theorem 1.

4.3. Discussions on the SBL Algorithm

In this subsection, we provide several additional discussions on the practical implementation and performance of the SBL Algorithm.

Practical Setting: Mixture of New and Mature Ads. So far we have assumed all ads are new and would generate some cold start rewards. However, in practice, new and mature ads are



(a) Average Regret (b) Average Regret Scaled by $t^{-2/3}(\log t)^{-1/3}$

Figure 3 Average Regret and Scaled Regret in the Simulation with $\alpha = 0.001$ and $\beta_j = 2b_j$

mixed together on the DSP to bid for user impressions. We now discuss how our model and SBL algorithm could be adapted to this practical setting with a mixture of new and mature ads. Note that mature ads can be viewed as new ads without any cold start value, which is positive only when the number of clicks is below the target αT. Therefore, we unify the representation of all ads on the DSP using a single parameter $\alpha \geq 0$, where $\alpha > 0$ for new ads and $\alpha = 0$ for mature ads. Define $[K']$ as the set of mature ads. We have the following dual formulation of the problem with mixed new and mature ads.

$$\min_{\lambda_j \in [0,\beta_j]} \quad \sum_{s \leq t-1} \max_{j \in [K] \cup [K']} \{\hat{c}_{ij}^s(\lambda_j + b_j)\} - \alpha T \sum_{j \in [K]} \lambda_j \tag{8}$$

It follows immediately from the complementary slackness condition that the solution to (8) satisfies that, for $j \in K'$, $\lambda_j = 0$. Therefore, to incorporate mature ads into the SBL algorithm without affecting its effectiveness, it suffices to include their bid prices into Step 1 of the algorithm.

Numerical Simulation. To numerically evaluate our algorithm, we build a simulation system of the DSP using the real data from Platform O. The data set contains one-week log records of 300 ads, including the impressions, clicks, conversions, viewer features, and bidding prices. To keep the simulation system clear, we focus on the CPC billing option and first-price auction. We build a ground truth model for the click-throughs of each ad via fitting a logistic regression model on historical impressions and clicks. We also embed another logistic regression model with randomly initialized parameters to generate pCTR in real-time. With this simulation system, we conduct two sets of numerical experiments.

In the first numerical experiment, we study a pure cold start problem with new adds only. We randomly select 100 new ads from the data set as well as 20,000 impressions to be allocated. The target number of clicks for each ad per one period is $\alpha = 0.001$, which is scaled in consistency with the cold start success criteria of 10 conversions in the three-day cold start horizon. We set the cold start reward coefficient $\beta_j = 2b_j$ for each ad j. We run the simulation for 50 times and compute

the average regret. The results are plotted in in Figure 3, which confirms our theoretical result (Theorem 1) that the regret of SBL algorithm is bounded by $O(t^{2/3}(\log t)^{1/3})$.

In the second numerical experiment, we include both new and mature ads in the simulation to better replicate the real-world DSP. In this experiment, we show that, with properly chosen cold start value parameters $\boldsymbol{\beta}$, our SBL algorithm indeed significantly increases the long-term platform revenue (Figure 8 in Appendix D). Interested readers are referred to Appendix D for more details of the simulation system.

5. Field Experiment Design and Algorithm Implementation

To demonstrate the practical value of our SBL algorithm, we conduct a two-sided randomized field experiment to causally test the impact of the algorithm on both the revenue and the cold start reward/success rate of the DSP. In this section, we first discuss our field setting and introduce our two-sided experiment design. Then we present the online implementation details of our algorithm.

5.1. Two-Sided Experiment Design

We collaborate with a large-scale online video-sharing platform (Platform O), where *in-feed* advertising contributes to a substantial share of its revenue. In-feed ad is the most common format of video ads in social media platforms, and is organized in a regulated form and intertwined with other organic content updates (In the video-sharing Platform O, in-feed ad is regulated in the video format). Compared with other ad formats such as banner ads, in-feed ads are more visually appealing to users and yield much higher revenues. Each user of Platform O can upload and watch videos thereon. As a user swipes up on the screen, a new organic video or an in-feed ad will be shown to the user. Platform O features interactive short videos (whose length is typically within 30 seconds) and in-feed ads, which is more likely to TikTok than to YouTube. Thanks to the interactive UI of the product, all in-feed ads on Platform O are in a short-video form with an "Ad" label, so a user can swipe up to skip an ad at any time. On YouTube, however, users have to watch a certain length of an ad before skipping it. If a user is interested in an ad, s/he may click the button that directs him/her to external sites such as AppStore to download the smartphone app and an e-commerce website for on-line shopping, etc. The user is converted if s/he finish the target action set by the advertiser such as downloading the app and purchasing the product. Figure 4 illustrates this process.

One may want to test the effectiveness of our algorithm by either randomly assigning new ads, or randomly assigning user views into treatment and control groups, as shown in Figure 5(a) and (b), respectively. However, both designs would violate the Stable Unit Treatment Value Assumption

[6] For protecting the platform's identity, we created the screenshots of an in-feed ad on another large online advertising platform in Figure 4, which has a similar interface as Platform O.

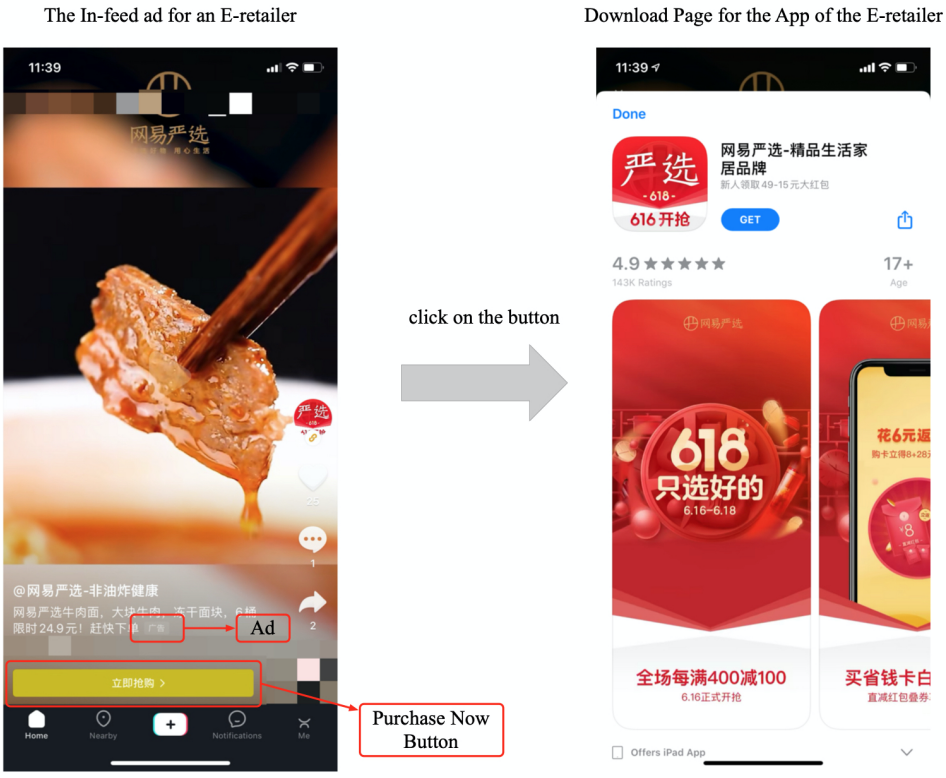


Figure 4 **Illustration of how ads are displayed to users** [6]

(SUTVA) (Imbens and Rubin 2015), thus causing biased estimates of the effect of the new algorithm (Johari et al. 2020). Blake and Coey (2014) empirically document that estimates from one-sided experiments in auctions with interference among their bidders could be biased by a factor of two. Specifically, if we randomly assign the new ads to treatment and control groups, after which we apply the SBL algorithm to all ads in the treatment group and the baseline cold start algorithm of the DSP (the system bidding prices generated by the PID controller; see Section 3) to those in the control group (see Figure 5(a)), the ads using our new algorithm will compete with those using the baseline algorithm on the same set of impressions, so the effect of the SBL Algorithm will be *overestimated*. Alternatively, one may conduct an experiment that randomizes over user impressions, in which impressions are randomly assigned to treatment and control groups each using different algorithms (see Figure 5(b)). See Schwartz et al. (2017) for an application of this experiment design in customer acquisition via display advertising. In Platform O's setting, however, the design in Figure 5(b) is also invalidated, again due to the violation of SUTVA: Both the SBL Algorithm and the baseline algorithm are applied to the same new ads, so the effect of SBL will spill over to the control group and eventually be *underestimated*. To illustrate the potential biases of the estimates from one-sided experiments in our ad cold start setting, we conduct some simulation studies to compare the estimation results of ad-side, user-side, and two-sided experiments (see

Appendix D.2 for the simulation details). Our simulation results (Table 6 in Appendix D.2) indicate that the ad-side experiment overestimates the cold start success rate by as much as 120%, whereas the user-side experiment underestimates this metric by 40%.

	Treatment New Ads	Control New Ads	Non-Experiment New Ads	Mature Ads
100% UV	Treatment Condition	Control Condition		

(a) Experiment with Ad Side Randomization

	100% New Ads	Mature Ads
Treatment UV	Treatment Condition	
Control UV	Control Condition	
Non-Experiment UV		

(b) Experiment with UV Side Randomization

Figure 5 Illustration of One-Side Experiments

To address the aforementioned SUTVA violation issue under one-sided experiments, we design a novel two-sided field experiment to evaluate the impact of our SBL algorithm. The experiment is conducted from May 23, 2020 to May 30, 2020, and the experiment design is illustrated in Figure 6. More specifically, we randomly assign 33% platform users' views into the treatment group (UT) and another 33% users' views (UVs) into the control group (UC). On the ad side, we randomly assign 20% of the new ads to the treatment group (AT) and 20% to the control group (AC). The salient feature of this design is that the treatment (resp. control) ads can only bid on the treatment (resp. control) UVs, but are not allowed to bid on control (resp. treatment) UVs. Through such a two-sided randomization design, the SUTVA condition is restored to the greatest extent we can. One underlying assumption is that the CTR and CVR distributions of the mature ads are not interfered by the cold start algorithm the DSP adopts.[7] Furthermore, it is assumed that on Platform O, the SBL algorithm does not affect how many ad views a user has during the experiment.[8] In the following, we will report our findings from the experiment to demonstrate the impact of our SBL algorithm.

[7] To verify this assumption, we sample 13,337 mature ads one day before the experiment and compare their empirical CTR before and during the experiment. The average CTR before the experiment is 13.11% with standard deviation 0.099, while the average CTR during the experiment is 13.19% with standard deviation 0.100. The p-value of the pairwise t-test is 0.284 and 0.481 for CTR and CVR, respectively, implying that our algorithm does not substantially change mature ads CTR and CVR.

[8] To test this assumption, we conduct a t-test of average ad impressions per user with p-value equal to 0.96. Hence, our algorithm does not change the number of user views significantly.

	20% Treatment New Ads	20% Control New Ads	60% Non-Experiment New Ads	Mature Ads
33% Treatment UV	B11	B12	B13	B14
33% Control UV	B21	B22	B23	B24
33% Non-Experiment UV	B31	B32	B33	B34

Figure 6 Illustration of the Two-Side Experiment Design

Note: 1. The new ads in cells B11 are bid with the SBL Algorithm (i.e., the shadow bids λ^* will be added towards the system bidding prices); 2. The new ads in cells (B21, B31, B12, B32) are forbidden to join the auction. 3. All other ads in uncolored cells join the auction following the system bidding prices without shadow bids.

5.2. Online Implementation of the Algorithm

We highlight a key advantage of our SBL algorithm that it can be easily adapted into the infrastructure of Platform O's DSP. Such convenience has enabled us to actually implement our SBL algorithm online. The actually implemented version of the algorithm is as follows:

Online Shadow Bidding with Learning (oSBL) Algorithm

Parameters: Set epoch schedule $0 = \tau_1 < \tau_2 < \cdots < \tau_m = T$ with fixed one-hour intervals; the cold start value coefficient $\beta_j = 2b_j$; and the conversion target $\alpha T = 10$ for new ads.

Initialization: $\lambda_j = 0$ for all j, $t = 0$, $m = 1$

For $t = 1, 2, \ldots$ **do**

Step 1: Observe the context i at round t. Choose top 150 ads (including new and mature ads, ranked by a pre-ranking model[9]), together with 15 randomly picked new ads, to join the auction.

Step 2: Get $\hat{c}_{ij}^t$, the estimate of pCTR×pCVR. Display the ad $a_t^* = \arg\max_{j\in[K_t]} \hat{c}_{ij}^t(b_{tj} + \lambda_j^{\tau_m})$, where b_{tj} is the system bidding price calculated by real-time PID system for ad j at period t. $[K_t]$ is the set of 165 ads who join the auction at time t.

Step 3: If $t = \tau_m$, update the history at period t $\mathcal{H}_t$, constructed by sampling 4% of the auctions in the past hour. Update $m = m + 1$, and $\lambda_j^{\tau_m}$ for each ad j by solving the following dual program,

$$\begin{aligned} \min \quad & \sum_{s\in[\mathcal{H}_t]} \max_{j\in[K_s]} \{\hat{c}_{ij}^s \lambda_j + \hat{c}_{ij}^s b_{sj}\} - \alpha|\mathcal{H}_t| \sum_{j\in[K_s]} \lambda_j \\ \text{s.t.} \quad & \lambda_j \in [0, \beta_j],\ \forall j \in [K],\ \lambda_j = 0,\ \forall j \in [K'] \end{aligned}$$

Step 4: Observe the outcome of ad a_t^*, and update the parameters in the neural network, which will estimate $\hat{c}_{ij}^s$ for all future periods $s \geq t + 1$. The advertiser will be charged based on the real-time system bidding price b_{tj}, but not the total bid $b_{tj} + \lambda_j^{\tau_m}$.

It is worth emphasizing that several aspects of the oSBL algorithm are different from the original SBL algorithm. First, the bid for each ad j in oSBL will follow the system bidding prices generated by the PID system, so the bid will change over time. As is clear from the oSBL algorithm, we can easily incorporate the real-time system bidding prices by adding the adaptive shadow bids produced by the dual solution in each epoch. Furthermore, the two-sided experiments (Figure 6)

[9] On Platform O's DSP, there are two stages before an ad enters the final auction: filtering and pre-ranking, both of which adopt deep neural network models to rule out the ads not suitable for the user impression.

can also be easily implemented online by adjusting the shadow bids according to which cell the ad-UV pair belongs to. Second, we set the fixed one-hour epoch schedule interval in oSBL. On one hand, this makes the pace of the algorithm consistent with other online systems of the DPS, such as the predictive models for pCTR and pCVR, and the PID controller. On the other hand, it alleviates the computational burden of the algorithm so that the shadow bids can be generated in a timely manner. Third, the exploration of the oSBL algorithm is to randomly add 15 new ads into the final auction for each impression, instead of the ϵ-greedy scheme proposed in the SBL algorithm. This adjustment is mainly driven by that we inherit the 10%-exploration heuristic that has already been implemented by Platform O's DSP. Making minimum changes to the online system of the DSP will ensure the robustness of our new algorithm. In fact, this exploration strategy is similar to the uniform exploration approach of MAB algorithms.

Fourth, when computing the shadow bids λ_j^* for each ad j, we sample 4% of the total auctions for user impressions. Such a downsampling approach could further reduce the computational burden of the oSBL algorithm. As a matter of fact, our algorithm could produce robust shadow bids even with a sampling rate of only 1%, as shown by our robustness check results in Appendix F. Finally, we remark that we set the cold start reward coefficient $\beta_j = 2b_j$ and the conversion target $\alpha T = 10$ mainly because of the business practice of Platform O's DSP. We have checked the robustness of our results for different conversion targets (see Figure 7). Furthermore, our numerical results, as presented in Appendix D, also substantiate that $\beta_j = 2b_j$ would produce the highest expected total long-term revenue of the platform (see Figure 8).

6. Field Experiment Results

In this section, we present the results of our two-sided field experiment. We document both the model-free and the regression-based results to demonstrate the value of our new algorithm.

6.1. Data and Randomization Check

We first check the randomization of the experiment. To confirm the success of our randomization in the two-sided experiment, we check the randomization on both the ad side and the UV side. For the ad side randomization check, we report the ad side randomization check results in Table 2 Panel A, where the numbers are re-scaled to protect the sensitive data. Table 2 Panel A shows that treatment and control ads in our sample were similar in bidding prices, the proportion of ads targeting iOS users, the proportion of ads targeting UI Version X, and the proportion of ads in various industries. We remark that these features are all submitted by the advertiser once s/he launches a new ad on the DSP and, therefore, are not affected by the algorithm of choice. Similarly, the UV side randomization check results are reported in Table 2 Panel B, where the numbers are also rescaled to protect the sensitive data. As we can see from Table 2 Panel B, treatment UVs

Table 2 Randomization Check of the Experiment

Panel A: Randomization Check on the Ad side				
		Treatment ads	Control ads	p-value of t-test
Statistics during the Experiment	Number of New Ads	34,605	34,076	
	Bidding Price	48.14	48.17	0.91
		(52.24)	(51.45)	
	Proportion of Ads for iOS Users	24.1%	24.2%	0.98
		(0.427)	(0.428)	
	Proportion of Ads for UI Version X	30.3%	28.3%	0.69
		(0.459)	(0.450)	
	Proportion of Ads in Game Industry	13.8%	13.7%	0.98
		(0.086)	(0.081)	
	Proportion of Ads in Education Industry	0.75%	0.67%	0.93
		(0.086)	(0.082)	
	Proportion of Ads in Finance Industry	1.75%	1.87%	0.93
		(0.131)	(0.135)	
Panel B: Randomization Check on the UV side				
		Treatment UV	Control UV	p-value of t-test
Statistics during the Experiment	Number of Users	197,460,792	197,401,621	
	Male Proportion	0.540	0.540	>0.99
		(0.491)	(0.491)	
	Average Revenue per User	0.95	0.95	>0.99
		(27.15)	(27.14)	
	Average Impressions per User	23.36	23.24	0.95
		(17900)	(17864)	
	Average Clicks per User	3.195	3.20	0.99
		(4455)	(4458)	
	Average Conversions per User	0.041	0.040	0.88
		(32.80)	(32.25)	

Note: Standard deviations in Panel A are clustered at the ad level and reported in the parentheses. Standard deviations in Panel B are clustered at the user level and reported in the parentheses. To protect sensitive data, the reported metrics are rescaled.

and control UVs generate similar revenues, ad impressions, ad clicks, and ad conversions per hour. The proportion of female users in the treatment group is also similar to that in the control group. We have thus confirmed that the treatment ads (resp. UVs) and control ads (resp. UVs) in our sample are comparable, implying that any difference between groups after the experiment started should be attributed to whether our new oSBL algorithm has been implemented.

6.2. Model-Free Results

We evaluate the performance of our proposed oSBL algorithm based on the following metrics:

1. *Cold Start Success Rate.* The cold start success rate is defined as the proportion of ads whose total number of conversions exceeds the conversion target, i.e., $\frac{\sum_{j=1}^{K}\mathbb{I}_{\{V_j\geq\alpha T\}}}{K}$, where K is the total number of new ads assigned to the respective experimental group. For Platform O, the cold start period of any ad is the first 3 days whereas the conversion target $\alpha T = 10$. One should note that new ads that arrive in the last three days do not pass the entire cold start period, but this will not affect the comparisons between the treatment and control groups.
2. *Cold Start Reward.* The cold start reward is clustered at the ad level, i.e., $\beta_j \min\{V_j, \alpha T\}$.
3. *Revenue.* This metric is clustered at the UV level for *all* (old and mature) the ads in different experiment groups on the DSP.

4. *Ratio between Real Cost and Target Cost.* This statistic is clustered at the ad level to evaluate the impact of our oSBL algorithm on the controllability of advertisers' cost. If this ratio is much bigger than 1, it implies that our new algorithm substantially increases the cost per conversion for the advertisers, which may cause them to complain or even leave the platform.

We summarize our model-free experimental results in Table 3. It is evident from Panel A of the table that our oSBL algorithm has substantially increased both the cold start success rate and the cold start reward of new ads by 61.62% and 47.41% (p-values ≤ 0.0001). Such improvements result from the shadow bids produced by oSBL that are added to the real-time system bidding prices, which also give rise to a 44.94% increase in ad impressions, a 35.34% increase in ad clicks, and a 59.67% increase in ad conversions (p-values ≤ 0.01). One may question whether the improvement in cold start success rate and cold start reward is at the cost of increased advertiser costs. Panel B of Table 3 addresses this question by examining the distribution of the relative gap between the real cost of an ad and its target cost gap (measured by $\frac{\text{Real Cost}}{\text{Target Cost}} - 1$). It shows that there is no significant difference between the distribution of the relative gap for ads in the treatment group and that for ads in the control group. More specifically, the treatment and control groups are similar in the proportion of new ads whose relative cost gap is in each of the following ranges [-30%,30%], [30%,100%], or $> 100\%$. The results show that the oSBL algorithm does not cost the advertisers more to increase the cold start success rate and the cold start reward of their new ads.

Although our oSBL algorithm significantly improves the cold start performance of new ads, it could decrease the impressions and conversions of mature ads and, as a consequence, reduces the total short-term revenue. Comparing the per-UV revenue between treatment and control groups on the UV side, Panel C of Table 3 shows that the oSBL algorithm will decrease the total revenue by 0.717% (with p-value less than 0.01). This small relative decrease in short-term revenue is within our expectation and is acceptable for Platform O. First, as demonstrated in our simulation, our proposed algorithm can successfully increase the thickness of the platform and give rise to higher long-term revenue. Second, our field experiments show that the oSBL algorithm can successfully balance the short-term revenue and the long-term cold start reward: The total objective function value is significantly increased by 0.147% with 20% new ads and 33% UVs in the treatment condition, as shown in Panel C of Table 3. On one hand, given the gigantic scale of Platform O, such an increase would imply a sizable hundred-million-dollar boost in the ad revenue per year in the long-run. On the other hand, because of the positive two-sided network effects between ads and users on Platform O, we would expect a much higher increase in the total value generated by our oSBL algorithm if applied to all the new ads and full user traffic. It remains an open problem and left as a future research direction to extrapolate the estimate of the SBL algorithm's effect based on 20% ads and 33% users to the entire population. Interested readers are referred to Appendix D.3

Table 3 The Effects of the Algorithm

Panel A: Effects on the Cold Start at Ad Level			
	Time Window: May 23, 2020 - May 30, 2020		
	Treatment		Control
	(1)		(2)
Number of Impressions	29,866 (268,139)		20,605 (232,143)
Relative effect size		44.94%****	
Number of Clicks	2,352 (24,088)		1,738 (30,780)
Relative effect size		35.34%**	
Number of Conversions	9.38 (108.93)		5.86 (68.67)
Relative effect size		59.67%****	
Observations	34,605		34,076
Cold Start Success Rate	0.0438 (0.204)		0.0271 (0.162)
Relative effect size		61.62%****	
Cold Start Reward	261.6 (958.1)		177.1 (930.5)
Relative effect size		47.71%****	
Observations	34,605		34,076
Panel B: The Effects of the Algorithm on Advertiser Costs			
$\frac{\text{Real Cost}}{\text{Target Cost}} - 1$	$-30\%\sim30\%$	$30\%\sim100\%$	$>100\%$
	(1)	(2)	(3)
Proportion of ads (Treatment Condition)	0.665	0.041	0.059
	(0.472)	(0.198)	(0.235)
Proportion of ads (Control Condition)	0.648	0.026	0.045
	(0.477)	(0.159)	(0.209)
p-value of t-test	0.74	0.45	0.58
Panel C: The Effects of the Algorithm on Revenue and the Objective Value			
	Time Window: May 23, 2020 - May 30, 2020		
	Treatment		Control
	(1)		(2)
Revenue Per User	1.439 (37.81)		1.448 (37.83)
Relative Effect Size of the Total Revenue		-0.717%**	
Observations	240,308,309		240,538,298
Total Objective Value	354,856,325		354,334,315
Relative effect size		0.147%****	

Note: Standard errors in both Panel A and Panel B are clustered at the ad level and reported in the parentheses. *p<0.1; **p<0.01; ***p<0.001; ****p<0.0001. To protect sensitive data, the impressions, clicks, and conversions data in Panel A are linearly scaled, as well as the reported metrics in Panel B and Panel C. The cutoff ranges $[-30\%, 30\%]$, $[30\%, 100\%]$, and $> 100\%$ are adopted in consistency with Platform O's business practice.

for some simulation studies of this problem. Finally, one should note that an advertiser does not need to pay for the shadow bid if his/her ad actually wins an auction and the user gets converted. Considering that the oSBL algorithm has significantly improved the cold start performances of new ads without a negative impact on cost controllability (see Panel B of Table 3), one may design a cost-sharing mechanism embedded in our oSBL algorithm under which the advertiser shares $\zeta\lambda_j^*$ and the DSP shares $(1-\zeta)\lambda_j^*$ of the entire optimal shadow bid λ_j^* for ad j ($\zeta \in (0,1)$). It would be an interesting future research direction to study how we could design such cost-sharing mechanism to improve the total welfare of both the platform and the advertisers.

The results above indicate that our new algorithm performs quite well with respect to the cold start success rate and cold start reward of new ads. One may wonder whether the effectiveness of our algorithm is robust with respect to the cold start criteria (10 conversions in the first 3 days). To demonstrate the robustness of our algorithm, we plot the experimental results of k-Cold-Start Success Rate[10] in Figure 7, where the absolute value of k-Cold-Start Success Rate has been rescaled

[10] This is a generalization of Cold Start Success Rate: The proportion of ads whose total number conversions exceed k in three days.

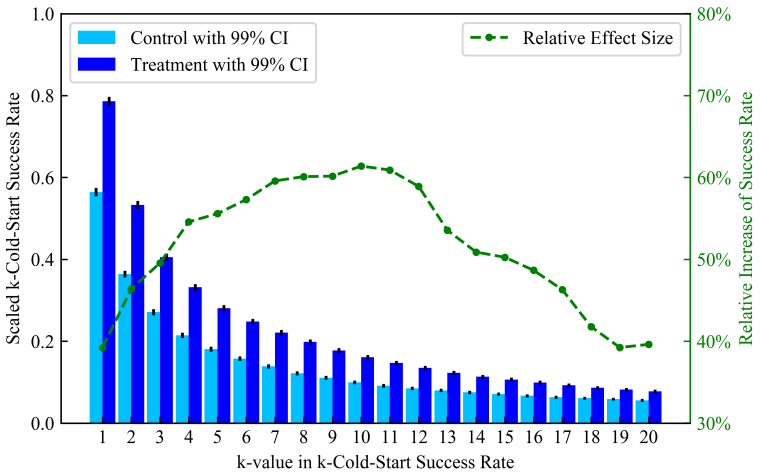


Figure 7 Effects of the Algorithm on the k-Cold-Start Success Rate

to protect the sensitive data. Figure 7 clearly shows that, irrespective of the choice of the conversion target, the oSBL algorithm causes substantial and robust improvement in the cold start success rate by 40% to 60%. This is a particularly important observation about the robustness of SBL algorithm regarding *model mis-specification*. We pick $\alpha T = 10$, conforming to the current business practice of Platform O, but this choice may not be the best one. Even with this sub-optimal choice, the oSBL algorithm still significantly boosts the success rate of cold start under various criteria.

Last but not least, we remark that our proposed SBL algorithm has substantially increased the prediction accuracy for the CTR of new ads. Specifically, our two-sided experiments show that the AUC of new ads CTR prediction in the treatment group is significantly larger than the one in the control condition by 7.48%, with the p-value of t-test being 0.017 (to protect the sensitive data, we only report the relative difference here).

6.3. Regression Based Results

In this section, we replicate our main results using the following linear regression specification which controls for the ad features to improve the efficiency of our estimators:

$$\text{Performance Indicator}_j = \alpha_0 + \alpha_1 \text{Treatment}_j + X_j + \epsilon \tag{9}$$

For the impact of the new algorithm on cold start reward and cold start success rate, Treatment_j is 1 if ad j is in the treatment group, otherwise 0; X_j is ad-specific features including the industry category (of the advertiser), bidding price, and some other UV-specific features. We first categorize UVs into different groups based on their age, gender, location, and device features. Then, we average over the UV features for each ad j and take this mean value as X_j. We also use the specification (9) to check the robustness of results on the revenue implications of our oSBL algorithm, where Treatment_j is 1 if user j is in the treatment group, otherwise 0; X_j is ad-specific features including

Table 4 Regression based Effects of the Algorithm

	Dependent Variable:			
	Cold Start Reward	Cold Start Success Rate	Revenue Per User	Objective Value
	(1)	(2)	(3)	(4)
Treatment−Control	73.0	0.0146	-0.0072	556,607
	(5.04)	(<0.001)	(<0.001)	
Relative Effect Size	41.22%****	53.87%****	-0.592%**	0.157%****
Model-Free Relative Effect Size	47.71%****	61.62%****	-0.717%**	0.147%****
Industry Fixed Effects	Yes	Yes	Yes	Yes
Bidding Price Effects	Yes	Yes	Yes	Yes
UV-specific Controls	Yes	Yes	Yes	Yes

Note: *p<0.1; **p<0.01; ***p<0.001; ****p<0.0001. Standard errors in column (1) and (2) are at the ad level and reported in the parentheses. Standard error in column (3) is at the user level.

the industry category (of the advertiser) and bidding price, averaged over all the ads displayed to each user j, and some other user-specific features, such as age, gender, location, and device features. For conciseness, we only report the three most important metrics of the platform, namely the cold start success rate and the cold start reward, as well as the revenue. The result of specification (9) is presented in Table 4. The regression-based results indicate that, after controlling for ad- and user-specific characteristics, our algorithm can significantly increase the cold start success rate by 41.22%, the cold start reward by 53.87%, and the total objective value by 0.157%, which are similar to the model-free results (see Section 6.2) in both directions and magnitudes.

7. Discussion and Conclusion

We close our paper by discussing several promising directions for future research. In Sections 3 and 5, we detail the cost control problem brought by complicated auction mechanisms and bidding/payment methods in a real DSP, which calls for future research for the cold start algorithm under real-time bidding. Furthermore, our cold start algorithm, in principle, can be embedded into a recommender system for general content as well. In this setting, ranking is a prominent data-driven decision. Integrating MAB algorithms and state-of-art ranking models such as Learning2Rank would be an interesting extension of our work. Finally, it is interesting to test other ways of conducting two-sided field experiments and quantify the biases that may be introduced by the violation of SUTVA in two-sided platforms.

References

Agarwal, Alekh, Daniel Hsu, Satyen Kale, John Langford, Lihong Li, Robert Schapire. 2014. Taming the monster: A fast and simple algorithm for contextual bandits. *International Conference on Machine Learning*. 1638–1646.

Agrawal, Shipra, Nikhil R Devanur, Lihong Li. 2016. An efficient algorithm for contextual bandits with knapsacks, and an extension to concave objectives. *Conference on Learning Theory*. 4–18.

Agrawal, Shipra, Zizhuo Wang, Yinyu Ye. 2014. A dynamic near-optimal algorithm for online linear programming. *Operations Research* **62**(4) 876–890.

Arora, Sanjeev, Simon S Du, Wei Hu, Zhiyuan Li, Russ R Salakhutdinov, Ruosong Wang. 2019. On exact computation with an infinitely wide neural net. *Advances in Neural Information Processing Systems*. 8139–8148.

Badanidiyuru, Ashwinkumar, Robert Kleinberg, Aleksandrs Slivkins. 2013. Bandits with knapsacks. *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*. IEEE, 207–216.

Balseiro, Santiago R, Omar Besbes, Gabriel Y Weintraub. 2015. Repeated auctions with budgets in ad exchanges: Approximations and design. *Management Science* **61**(4) 864–884.

Bastani, Hamsa, David Simchi-Levi, Ruihao Zhu. 2019. Meta dynamic pricing: Learning across experiments. *arXiv preprint arXiv:1902.10918* .

Bharadwaj, Vijay, Peiji Chen, Wenjing Ma, Chandrashekhar Nagarajan, John Tomlin, Sergei Vassilvitskii, Erik Vee, Jian Yang. 2012. Shale: an efficient algorithm for allocation of guaranteed display advertising. *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. 1195–1203.

Bimpikis, Kostas, Wedad J Elmaghraby, Ken Moon, Wenchang Zhang. 2020. Managing market thickness in online business-to-business markets. *Management Science* .

Blake, Thomas, Dominic Coey. 2014. Why marketplace experimentation is harder than it seems: The role of test-control interference. *Proceedings of the fifteenth ACM conference on Economics and computation*. 567–582.

Boucheron, Stéphane, Gábor Lugosi, Pascal Massart. 2013. *Concentration inequalities: A nonasymptotic theory of independence*. Oxford university press.

Caldentey, René, Gustavo Vulcano. 2007. Online auction and list price revenue management. *Management Science* **53**(5) 795–813.

Cao, Yuan, Quanquan Gu. 2019. Generalization bounds of stochastic gradient descent for wide and deep neural networks. *Advances in Neural Information Processing Systems*. 10836–10846.

Chen, Boxiao, Xiuli Chao, Hyun-Soo Ahn. 2019. Coordinating pricing and inventory replenishment with nonparametric demand learning. *Operations Research* **67**(4) 1035–1052.

Chen, Ningyuan, Guillermo Gallego. 2018. A primal-dual learning algorithm for personalized dynamic pricing with an inventory constraint. *Available at SSRN 3301153* .

Chen, Weidong, Cong Shi, Izak Duenyas. 2020. Optimal learning algorithms for stochastic inventory systems with random capacities. *Production and Operations Management* .

Chizat, Lenaic, Edouard Oyallon, Francis Bach. 2019. On lazy training in differentiable programming. *Advances in Neural Information Processing Systems*. 2937–2947.

Choi, Hana, Carl F Mela, Santiago R Balseiro, Adam Leary. 2020. Online display advertising markets: A literature review and future directions. *Information Systems Research* .

Chu, Wei, Lihong Li, Lev Reyzin, Robert Schapire. 2011. Contextual bandits with linear payoff functions. *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. 208–214.

Cui, Ruomeng, Jun Li, Dennis Zhang. 2019a. Discrimination with incomplete information in the sharing economy: Evidence from field experiments on airbnb. *Management Science* Forthcoming.

Cui, Ruomeng, Dennis J Zhang, Achal Bassamboo. 2019b. Learning from inventory availability information: Evidence from field experiments on amazon. *Management Science* **65**(3) 1216–1235.

Dave, Kushal, Vasudeva Varma. 2012. Identifying microblogs for targeted contextual advertising. *Sixth International AAAI Conference on Weblogs and Social Media*.

Dave, Kushal, Vasudeva Varma. 2014. Computational advertising: Techniques for targeting relevant ads. *Foundations and Trends in Information Retrieval* **8**(4–5) 263–418.

Devanur, Nikhil R, Thomas P Hayes. 2009. The adwords problem: online keyword matching with budgeted bidders under random permutations. *Proceedings of the 10th ACM conference on Electronic commerce*. 71–78.

Dudík, Miroslav, John Langford, Lihong Li. 2011. Doubly robust policy evaluation and learning. *International Conference on Machine Learning*.

Feldman, Jake, Dennis J Zhang, Xiaofei Liu, Nannan Zhang. 2018. Taking asssortment optimization from theory to practice: Evidence from large field experiments on alibaba Working Paper.

Ferreira, Kris Johnson, David Simchi-Levi, He Wang. 2018. Online network revenue management using thompson sampling. *Operations research* **66**(6) 1586–1602.

Fisher, Marshall, Santiago Gallino, Jun Li. 2018. Competition-based dynamic pricing in online retailing: A methodology validated with field experiments. *Management science* **64**(6) 2496–2514.

Foster, Dylan, Alekh Agarwal, Miroslav Dudik, Haipeng Luo, Robert Schapire. 2018. Practical contextual bandits with regression oracles. *Proceedings of Machine Learning Research* **80**.

Foster, Dylan J, Alexander Rakhlin. 2020. Beyond ucb: Optimal and efficient contextual bandits with regression oracles. *arXiv preprint arXiv:2002.04926* .

Gallego, Guillermo, Garrett Van Ryzin. 1994. Optimal dynamic pricing of inventories with stochastic demand over finite horizons. *Management science* **40**(8) 999–1020.

Glaeser, Chloe Kim, Marshall Fisher, Xuanming Su. 2019. Optimal retail location: Empirical methodology and application to practice. *Manufacturing & Service Operations Management* **21**(1) 86–102.

Golrezaei, Negin, Adel Javanmard, Vahab Mirrokni. 2019. Dynamic incentive-aware learning: Robust pricing in contextual auctions. *Advances in Neural Information Processing Systems*. 9759–9769.

Golrezaei, Negin, Hamid Nazerzadeh, Paat Rusmevichientong. 2014. Real-time optimization of personalized assortments. *Management Science* **60**(6) 1532–1551.

Ha-Thuc, Viet, Avishek Dutta, Ren Mao, Matthew Wood, Yunli Liu. 2020. A counterfactual framework for seller-side a/b testing on marketplaces. *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2288–2296.

Hastie, Trevor, Robert Tibshirani, Martin Wainwright. 2015. *Statistical learning with sparsity: the lasso and generalizations*. CRC press.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, Jian Sun. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *Proceedings of the IEEE international conference on computer vision*. 1026–1034.

He, Pu, Fanyin Zheng, Elena Belavina, Karan Girotra. 2020. Customer preference and station network in the london bike-share system. *Management Science* .

Hojjat, Ali, John Turner, Suleyman Cetintas, Jian Yang. 2017. A unified framework for the scheduling of guaranteed targeted display advertising under reach and frequency requirements. *Operations Research* **65**(2) 289–313.

Hsu, Daniel, Sham M. Kakade, Tong Zhang. 2014. Random design analysis of ridge regression. *Foundations of Computational Mathematics* **14**(3) 569–600.

Imbens, Guido W, Donald B Rubin. 2015. *Causal inference in statistics, social, and biomedical sciences*. Cambridge University Press.

Jacot, Arthur, Franck Gabriel, Clément Hongler. 2018. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*. 8571–8580.

Johari, Ramesh, Hannah Li, Gabriel Weintraub. 2020. Experimental design in two-sided platforms: An analysis of bias. *arXiv preprint arXiv:2002.05670* .

Langford, John, Tong Zhang. 2007. The epoch-greedy algorithm for contextual multi-armed bandits. *Proceedings of the 20th International Conference on Neural Information Processing Systems*. Citeseer, 817–824.

Li, Xiaocheng, Chunlin Sun, Yinyu Ye. 2020. Simple and fast algorithm for binary integer and online linear programming. *arXiv preprint arXiv:2003.02513* .

Nambiar, Mila, David Simchi-Levi, He Wang. 2019. Dynamic learning and pricing with model misspecification. *Management Science* **65**(11) 4980–5000.

Riquelme, Carlos, George Tucker, Jasper Snoek. 2018. Deep bayesian bandits showdown: An empirical comparison of bayesian deep networks for thompson sampling. *arXiv preprint arXiv:1802.09127* .

Schwartz, Eric M, Eric T Bradlow, Peter S Fader. 2017. Customer acquisition via display advertising using multi-armed bandit experiments. *Marketing Science* **36**(4) 500–522.

Sutton, Richard S, Andrew G Barto. 2018. *Reinforcement learning: An introduction*.

Terwiesch, Christian, Marcelo Olivares, Bradley R Staats, Vishal Gaur. 2019. A review of empirical operations management over the last two decades. *Manufacturing & Service Operations Management* .

Valko, Michal, Nathaniel Korda, Rémi Munos, Ilias Flaounas, Nelo Cristianini. 2013. Finite-time analysis of kernelised contextual bandits. *arXiv preprint arXiv:1309.6869* .

Wager, Stefan, Guenther Walther. 2015. Adaptive concentration of regression trees, with application to random forests. *arXiv preprint arXiv:1503.06388* .

Yang, Xun, Yasong Li, Hao Wang, Di Wu, Qing Tan, Jian Xu, Kun Gai. 2019. Bid optimization by multivariable control in display advertising. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1966–1974.

Zeng, Zhiyu, Hengchen Dai, Dennis Zhang, Zuo-Jun Max Shen, Zhiwei Xu, Heng Zhang, Renyu Philip Zhang. 2020. Social nudges boost productivity on onlineplatforms: Evidence from field experiments. *Available at SSRN 3611571* .

Zhang, Dennis J, Hengchen Dai, Lingxiu Dong, Fangfang Qi, Nannan Zhang, Xiaofei Liu, Zhongyi Liu, Jiang Yang. 2020. The long-term and spillover effects of price promotions on retailing platforms: Evidence from a large randomized experiment on alibaba. *Management Science* **66**(6) 2589–2609.

Zhang, Heng, Paat Rusmevichientong, Huseyin Topaloglu. 2018. Multiproduct pricing under the generalized extreme value models with homogeneous price sensitivity parameters. *Operations Research* **66**(6) 1559–1570.

Zhang, Weinan, Yifei Rong, Jun Wang, Tianchi Zhu, Xiaofan Wang. 2016. Feedback control of real-time display advertising. *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*. 407–416.

Zhao, Wayne Xin, Sui Li, Yulan He, Edward Y Chang, Ji-Rong Wen, Xiaoming Li. 2015. Connecting social media to e-commerce: Cold-start product recommendation using microblogging information. *IEEE Transactions on Knowledge and Data Engineering* **28**(5) 1147–1159.

Zhou, Dongruo, Lihong Li, Quanquan Gu. 2020. Neural contextual bandits with ucb-based exploration. *International Conference on Machine Learning*.

Zhou, Guorui, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, Kun Gai. 2018. Deep interest network for click-through rate prediction. *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1059–1068.

Online Appendices

Appendix A: Table of Notations

Table 5 Table of Notations

Notations in the Allocation Model and SBL Algorithm	
Notation	Description
K	The number of ads
$A=\{1,2,\ldots,K\}$	The set of ads
T	The total number of the user views, namely, ad impressions
X	The finite or countably infinite context set
$x_t \in X$	The context associated with user view t
$m=\lvert X\rvert$	The cardinality of the context set X
$a_t \in A$	The ad which is chosen to be displayed to the user view t
$\boldsymbol{v}_t(a_t) \in \{0,1\}^K$	The K-dimensional binary vector representing whether each ad is clicked
$\mathcal{D}_X$	The distribution over X in the stochastic bandit setting
$\mathcal{D}$	The distribution of $(x_t, \boldsymbol{v}_t(a_t))$ over $X \times \{0,1\}^K$ in the stochastic bandit setting
$c_{ij}=\mathbb{E}[v_{tj}(a_t)\lvert x_t=i]$	The CTR of ad j for user view t
$\boldsymbol{V} := \sum_{t=1}^{T} \boldsymbol{v}_t(a_t)$	The accumulated click-through vector
$b_j \in [0,1]$	The bid per click of ad $j \in A$
$\beta_j \in (0,1]$	The cold start value per click of ad $j \in A$
$\alpha \in (0,1)$	The target click per round
$\Gamma(\boldsymbol{V})$	The objective value
$\mathcal{H}_t=\bigcup_{s=1,\ldots,t-1}\{(x_s,a_s,\boldsymbol{v}_s(a_s))\}$	The history update to round t
$\Delta_A=\{\boldsymbol{y}\in\mathbb{R}^{\lvert A\rvert}: y_j\geq 0, \forall j\in A, \sum_{j\in A} y_j \leq 1\}$	The distribution over arms which defines the feasible ad allocation plan
π	The policy mapping from $\mathcal{H}_t$ to Δ_A
$\hat{c}_{tj}$	The empirically estimated CTR based on $\mathcal{H}_t$ at round t of ad j
$\hat{c}^t_{ij}$	The predicted CTR based on $\mathcal{H}_t$ of ad j under context i
$\boldsymbol{\lambda}^t$	The empirically optimal shadow bidding prices at round t
p_i	The probability that context $i \in X$ occurs
$\hat{p}^t_i$	The empirically estimated probability that context $i \in X$ occurs at round t
τ	The epoch schedule to update λ
$f(x)=O(g(x))$	There exists a positive constant c such that $f(x)\leq c\cdot\lvert g(x)\rvert$ for sufficiently large x
$f(x)=\tilde{O}(g(x))$	There exists a positive constant c such that $f(x)\leq c\cdot\lvert g(x)\rvert\cdot\log^k(\lvert g(x)\rvert)$ for some $k>0$ and sufficiently large x
$f(x)=\Omega(g(x))$	There exists a positive constant c such that $f(x)\geq c\cdot\lvert g(x)\rvert$ for sufficiently large x
$f(x)=\Theta(g(x))$	$\lvert f(x)\rvert/\lvert g(x)\rvert$ converges to 1 as x goes to infinity.
Notations in the Neural Network Prediction Oracle	
Notation	Description
$\mathcal{X}$	The set of functions $(X\times A\mapsto[0,1])$ to estimate CTR
$h(x_i,a_j), h_j(x_i)$	The ground truth CTR function such that $h(x_i,a_j)=c_{ij}$
w_0	The dimension of the context $x_t \in \mathbb{R}^{w_0}$
w	The number of hidden nodes of the neural network
L	The number of hidden layers of the neural network
d	The effective dimension of the regressor
$\theta \in \mathbb{R}^d$	The coefficients of parameters in the neural network
$\theta_0 \in \mathbb{R}^d$	The initialized coefficients of parameters in the neural network
$\theta^t \in \mathbb{R}^d$	The updated coefficients of parameters in the neural network at round t
$H_j(x_i,\theta)$	The output of the neural network given the coefficient θ, context i, and ad j
$\theta^* \in \mathbb{R}^d$	The coefficients such that $c_{ij}=\langle\nabla_\theta H_j(x_i,\theta_0), \theta^*-\theta_0\rangle$
λ_0	The regularization parameter in training the neural network
η	The step size in training the neural network
U	The number of descent steps in training the neural network
I_d	The identity matrix of dimension d
$\boldsymbol{H}$	The neural tangent kernel matrix defined by Zhou et al. (2020)
γ	The scalar such that $\boldsymbol{H}\succeq\gamma I$, where $M_1\succeq M_2$ refers to that M_1-M_2 is semi-positive-definite
Notations in Proofs	
Notation	Description
N^t_j	The set of contexts i for which $j\in\arg\max_{j'}\hat{c}^t_{ij'}(b_{j'}+\lambda_{j'})$
$\mathcal{H}^t_j$	The set of user views where ad j is displayed before the round t
$n^t_j=\lvert\mathcal{H}^t_j\rvert$	The cardinality of the set $\mathcal{H}^t_j$, i.e, the number of displays of ad j before round t
$g_{ij}=\nabla_\theta H_j(x_i,\theta_0)$	The gradients of the function $H_j(x_i,\theta_0)$
$\hat{\gamma}^t_j=\sum_{i\in N^t_j} p_i\hat{c}^t_{ij}$	The empirical estimated probability of click-through for ad j at round t
$\gamma^t_j=\sum_{i\in N^t_j} p_i c^t_{ij}$	The expected probability of click-through for ad j at round t
$n(j)$	The number of times that ad j is clicked over the whole horizon

Appendix B: Supporting Arguments for Regret Analysis

We devote this section to the proof of Theorem 1. Supporting analysis for justifying the prediction oracle assumption (Assumption 2) for the case of neural networks can be found in Appendix G. Before presenting the full-fledged proof of Theorem 1, we first give the proof of Lemma 1, followed by some preliminaries.

B.1. Proof of Lemma 1.

Let $\boldsymbol{y}^*$ denote the optimal solution of the optimization model in Lemma 1. Consider an arbitrary policy, we have the following observation:

$$\begin{aligned}\frac{1}{T}\cdot\mathbb{E}_{\mathcal{D}^T,\pi}\left[\Gamma(\boldsymbol{V})\right] &= \frac{1}{T}\mathbb{E}_{\mathcal{D}^T,\pi}\left[\sum_{j=1}^K b_j\sum_{t=1}^T v_{tj}+\sum_{j=1}^K\beta_j\min\left\{\sum_{t=1}^T v_{tj},\alpha T\right\}\right]\\ &= \mathbb{E}_{\mathcal{D}^T,\pi}\left[\sum_{j=1}^K b_j\sum_{t=1}^T v_{tj}/T+\sum_{j=1}^K\beta_j\min\left\{\sum_{t=1}^T v_{tj}/T,\alpha\right\}\right]\\ &\le \sum_{j=1}^K b_j\mathbb{E}_{\mathcal{D}^T,\pi}\left[\sum_{t=1}^T v_{tj}/T\right]+\sum_{j=1}^K\beta_j\min\left\{\mathbb{E}_{\mathcal{D}^T,\pi}\left[\sum_{t=1}^T v_{tj}/T\right],\alpha\right\},\end{aligned}$$

in which the inequality follows from the Jensen's inequality.

Let us use $\mathcal{A}_{tj}$ to denote the event that ad j is displayed for user t and $\mathcal{D}_{ij}$ to denote the distribution that ad j is clicked when the context is i and ad j is displayed. It then follows that

$$\begin{aligned}&\mathbb{E}_{\mathcal{D}^T,\pi}\left[\sum_{t=1}^T v_{tj}/T\right]=\sum_{t=1}^T\mathbb{E}_{i\sim\mathcal{D}_X}[\mathbb{E}_{\mathcal{H}^t,\pi,\mathcal{D}_{ij}}[v_{tj}|x_t=i]/T]=\sum_{t=1}^T\mathbb{E}_{i\sim\mathcal{D}_X}[\mathbb{P}_\pi(\mathcal{A}_{tj}|x_t=i)/T\cdot c_{ij}]\\ =&\mathbb{E}_{i\sim\mathcal{D}_X}\left[\sum_{t=1}^T\mathbb{P}_\pi(\mathcal{A}_{tj}|x_t=i)/T\cdot c_{ij}\right]=\mathbb{E}_{i\sim\mathcal{D}_X}[c_{ij}y_{ij}^\pi],\end{aligned}$$

in which we define $y_{ij}^\pi=\sum_{t=1}^T\mathbb{P}_\pi(\mathcal{A}_{tj}|x_t=i)/T$. Note that for any fixed i, it must be that $\sum_{j=1}^K y_{ij}^\pi=1$. Therefore,

$$\begin{aligned}&\frac{1}{T}\cdot\mathbb{E}_{\mathcal{D}^T,\pi}\left[\Gamma(\boldsymbol{V})\right]\le\sum_{j=1}^K b_j\mathbb{E}_{i\sim\mathcal{D}_X}[c_{ij}y_{ij}^\pi]+\sum_{j=1}^K\beta_j\min\left\{\mathbb{E}_{i\sim\mathcal{D}_X}[c_{ij}y_{ij}^\pi],\alpha\right\},\\ &\le\sum_{j=1}^K b_j\mathbb{E}_{i\sim\mathcal{D}_X}\left[c_{ij}y_{ij}^*\right]+\sum_{j=1}^K\beta_j\min\left\{\mathbb{E}_{i\sim\mathcal{D}_X}[c_{ij}y_{ij}^*],\alpha\right\}=\mathsf{OPT}.\end{aligned}$$

This concludes the proof. □

B.2. Preliminaries for Regret Analysis

We first make several additional assumptions in our proof, purely for the ease and clarity of exposition. First of all, instead of solving

$$\mathsf{OPT}^t=\min_{0\le\lambda_j\le\beta_j}\sum_{i\in X}\hat{p}_i^t\max_{j=1,2,\ldots,K}\left(\hat{c}_{ij}^t(b_j+\lambda_j)\right)+\alpha\sum_{j=1}^K(\beta_j-\lambda_j),\tag{10}$$

in the analysis, we assume that we solve

$$\mathsf{OPT}^t=\min_{0\le\lambda_j\le\beta_j}\sum_{i\in X}p_i\max_{j=1,2,\ldots,K}\left(\hat{c}_{ij}^t(b_j+\lambda_j)\right)+\alpha\sum_{j=1}^K(\beta_j-\lambda_j).\tag{11}$$

That is, we assume that we observe p_i for any $i\in X$, instead of only having access to its empirical estimate. In the meanwhile, we replace Assumption 1 with $p_i\hat{c}_{ij}\le O(T^{-\frac{1}{3}}(\log T)^{\frac{1}{3}}K^{-\frac{5}{3}})$ for all $i\in X$. This is without

loss of generality, because all the argument presented in this section still follows without this assumption. Indeed, with McDiarmid's inequality and the bound of Rademacher complexity term (Boucheron et al. 2013) with countably many contexts (i.e., $X = \mathbb{Z}^+ := \{1, 2, 3, ...\}$), we can still uniformly bound the error of the empirical probability estimate $\hat{p}_i^t$. Specifically, for any context $i \in X$, with probability at least $1 - t^{-4}$, we have $|\hat{p}_i^t - p_i| \le O(\sqrt{\log t / t})$, where t is the total number of occurrences for context i. As a result, this introduces a lower order error than $\tilde{O}(t^{-1/3})$, which can be ignored for our regret analysis. We will discuss more about this point at the end of the proof of Theorem 1 (see Appendix B.3).

Second, we assume that, when we solve (11), the inputs are in a *general position*. In other words, for any shadow bidding prices $\boldsymbol{\lambda}$ and round t when deciding which ad to display given a context, namely, $\left|\left\{i : |\arg\max_k \{\hat{c}_{ik}^t(\lambda_k + b_k)\}| > 1\right\}\right| \le K$. This assumption is introduced is to avoid too many ties in Step 1 of the SBL algorithm, thus bounding the gap between primal and dual solutions in a lower order compared to the total regret. Similar assumptions are also made in the online matching and online linear program literature, e.g., Devanur and Hayes (2009), Agrawal et al. (2014). As argued by Devanur and Hayes (2009), when the general position assumption does not hold, an infinitesimal permutation ξ_{ij} can be added to each $\hat{c}_{ij}^t$ without affecting much of the objective function value for any $\boldsymbol{\lambda}$, where ξ_{ij} is chosen independently and uniformly at random from a tiny interval $[-\varepsilon, \varepsilon]$ with ε being arbitrarily small. Hence, it is without loss of generality to assume the total number of ties is bounded by K with probability one. As will be clear in the proof of Lemma 2 below, the general position assumption helps us bound the total number of entries for the allocation decision constructed from the dual that are different from the primal solution by $O(K^2)$.

Lastly, we solve the dual program (7) in every exploitation round (i.e., $\tau_m - \tau_{m-1} = 1$). As will be shown in the Proof of Theorem 1 (see Appendix B.3), this assumption is made without loss of generality, because they will only result in an additional regret of a lower order than $\tilde{O}(T^{\frac{2}{3}})$.

Next, some definitions and notations are in order. Throughout the proof of Theorem 1, we define the reward process (for any policy) $\{r(x_t, a_t)\}_{t=1}^T$, where the reward at round t is denoted by:

$$r(x_t, a_t) = \begin{cases} 0, & \text{if } \ell_t(a_t) = 0, \\ b_{a_t}, & \text{if } \sum_{s=1}^{t-1} \ell_s(a_t) \ge \alpha T \text{ and } \ell_t(a_t) = 1, \\ b_{a_t} + \beta_{a_t}, & \text{if } \sum_{s=1}^{t-1} \ell_s(a_t) < \alpha T \text{ and } \ell_t(a_t) = 1, \end{cases}$$

where $\ell_t(a) = 1$ if and only if $a = a_t$ and a click-through occurs. Notice that the objective value is given by $\Gamma(\boldsymbol{V}) = \sum_{t=1}^T r(x_t, a_t)$. By Lemma 1, the expected reward satisfies $\mathbb{E}[\sum_{t=1}^T r(x_t, a_t)] \le T \cdot \mathsf{OPT}$. We note that $r(\cdot, \cdot)$ corresponds to the real reward collection process, which is hard to work with, because it depends on the click-through history of each ad so far. To overcome this challenge, we instead work with an auxiliary reward process, defined by

$$\tau(x_t, a_t) = \begin{cases} 0, & \text{if } \ell_t(a_t) = 0, \\ b_{a_t} + \beta_{a_t}, & \text{if } \ell_t(a_t) = 1, \end{cases}$$

Note $\tau(x_t, a_t) - r(x_t, a_t) = \beta_{a_t}$ when $\sum_{s=1}^{t-1} \ell_s(a_t) \ge \alpha T$ and a click-through occurs in round t. Otherwise, $\tau(x_t, a_t) = r(x_t, a_t)$.

Furthermore, we define N_j^t as the set of contexts i for which ad j is chosen to be pulled, with tie-breaking resolved, in round t. Thus, if $i \in N_j^t$, it holds that $j \in \arg\max_{j'} \hat{c}_{ij'}^t(b_{j'} + \lambda_{j'})$. Let $\boldsymbol{\lambda}^t$ be the optimal dual solution and $\boldsymbol{y}^t$ be the resulting (feasible integer) primal solution constructed by the SBL algorithm. Hence, $N_j^t = \{i : y_{ij}^t = 1\}$. We also define $(\boldsymbol{y}^*, \boldsymbol{u}^*)$ as the optimal solution in the primal space for round t.

Before the formally presenting the regret analysis, we also need to bound the gap induced by the tie-breaking problem in the SBL algorithm. Clearly, complimentary slackness implies that if $y^*_{i\ell} > 0$, then $\ell \in \arg\max_j \hat{c}^t_{ij}(b_j + \lambda^{\tau_m}_j)$. Hence, if $\arg\max_j \hat{c}^t_{ij}(b_j + \lambda^{\tau_m}_j)$ returns a unique solution ℓ, then the optimal primal solution and our constructed one are the same. Because we use the dual-based solution to make the ad allocation decision in the primal space, the tie-breaking in Step 2 of the SBL algorithm can induce the difference between empirically optimal objective value and the objective value by the dual-based allocation. One may expect that the solution to (11) still yields a good performance due to complimentary slackness, the general position assumption and Assumption 1. Formally, the following lemma holds.

LEMMA 2 **(Approximate Complimentary Slackness)**. *Under Assumption 1 and the SBL algorithm, there exist a family of non-negative constants* $\{\eta_j \geq 0 : j \in A\}$ *with* $\sum_{j\in A}\eta_j \leq O(T^{-\frac{1}{3}}(\log T)^{\frac{1}{3}}K^{\frac{1}{3}})$*, such that the following approximate complementary slackness condition holds for each* $j \in A$*:*

(i) If $\lambda^t_j \in [0, \beta_j)$*, we have* $\sum_{i\in N^t_j} p_i \hat{c}^t_{ij} \geq \alpha - \eta_j$*.*

(ii) If $\lambda^t_j \in (0, \beta_j]$*, we have* $\sum_{i\in N^t_j} p_i \hat{c}^t_{ij} \leq \alpha + \eta_j$*.*

Proof of Lemma 2.

To analyze the non-smooth convex dual (7), we first write down the corresponding primal linear program (12) and its dual (13) as follows,

$$\begin{aligned} \text{(Primal)}\quad \mathsf{OPT}^t = \max_{y_{ij}\geq 0} \quad & \sum_{i\in X}\sum_{j=1}^{K} p_i \hat{c}^t_{ij} b_j y_{ij} + \sum_{j=1}^{K}\beta_j(\alpha - u_j) \\ \text{s.t.} \quad & \sum_{j=1}^{K} y_{ij} \leq 1,\ \forall i \in X,\ \sum_{i\in X} p_i \hat{c}^t_{ij} y_{ij} + u_j \geq \alpha,\ \forall j \leq K, \end{aligned} \tag{12}$$

and,

$$\begin{aligned} \text{(Dual)}\quad \mathsf{OPT}^t = \min_{\lambda_j\geq 0} \quad & \sum_{i\in X} p_i\mu_i + \alpha\sum_{j=1}^{K}(\beta_j - \lambda_j) \\ \text{s.t.} \quad & \lambda_j \leq \beta_j\ \forall j \leq K \\ & \mu_i - \hat{c}^t_{ij}\lambda_j \geq \hat{c}^t_{ij} b_j\ \forall i \in X,\ \forall j \leq K. \end{aligned} \tag{13}$$

Clearly, the following complimentary slackness conditions

$$\lambda^t_j(\alpha - u^*_j - \sum_{i\in X} p_i \hat{c}^t_{ij} y^*_{ij}) = 0,\ u^*_j(\lambda^t_j - \beta_j) = 0,\ \text{and}\ y^*_{ij}(\mu_i - \hat{c}^t_{ij}\lambda^t_j - \hat{c}^t_{ij} b_j) = 0,$$

hold for any $i \in X$ and $j \leq m$. To highlight the intuition, let us first consider the case in which there is no tie in $\arg\max_{j'} \hat{c}^t_{ij'}(b_{j'} + \lambda_{j'})$ for any $i \in X$. Notice that if $\ell \notin \arg\max_{j'} \hat{c}^t_{ij'}(b_{j'} + \lambda_{j'})$, then the complementary condition implies that $y^*_{i\ell} = 0$. Since the primal program is increasing in $\boldsymbol{y}$, it must be that $y^*_{ij} = 1$ if $j = \arg\max_{j'} \hat{c}^t_{ij'}(b_{j'} + \lambda_{j'})$. In this case, $i \in N^t_j$ if and only if $y^*_{ij} = 1$, and $y^*_{ij} = 0$ otherwise . If $\lambda^t_j < \beta_j$, then $u^*_j = 0$. As a result, $\sum_{i\in N^t_j} p_i \hat{c}^t_{ij} = \sum_{i\in N^t_j} p_i \hat{c}^t_{ij} y^*_{ij} = \sum_{i\in X} p_i \hat{c}^t_{ij} y^*_{ij} + u^*_j \geq \alpha$. Similarly, if $\lambda^t_j > 0$, then $\alpha - u^*_j - \sum_{i\in X} p_i \hat{c}^t_{ij} y^*_{ij} = 0$, which implies that $\sum_{i\in N^t_j} p_i \hat{c}^t_{ij} \leq \alpha$. So, if there is no tie, the lemma holds true with $\eta_j = 0$, $\forall j \in A$.

Next, we consider the general case. By the general position assumption, there are at most K ties and, thus, the tie breaking introduces at most K^2 different entries from a primal optimal allocation $\boldsymbol{y}^*$ to obtain

the dual-based solution $\boldsymbol{y}^t$. This is because, with an argument similar to the one in the previous argument, for $i \in X$ such that there is no tie, it must be $\boldsymbol{y}^t_{i\cdot} = \boldsymbol{y}^*_{i\cdot}$ and all entries in $\boldsymbol{y}^t_{i\cdot}$ belongs to the set $\{0,1\}$. Hence,

$$\sum_{j=1}^{K}\left|\sum_{i\in N_j^t} p_i\hat{c}_{ij}^t - \sum_{i\in X} p_i\hat{c}_{ij}^t y_{ij}^*\right| = \sum_{j=1}^{K}\left|\sum_{i\in X} p_i\hat{c}_{ij}^t y_{ij}^t - \sum_{i\in X} p_i\hat{c}_{ij}^t y_{ij}^*\right|$$
$$\leq \sum_{j=1}^{K}\sum_{i\in X} p_i\hat{c}_{ij}^t\left|y_{ij}^t - y_{ij}^*\right| \leq O(K^2(T^{-\frac{1}{3}}(\log T)^{\frac{1}{3}}K^{-\frac{5}{3}})) = O(T^{-\frac{1}{3}}(\log T)^{\frac{1}{3}}K^{\frac{1}{3}}),$$

in which the first inequality follows from the definition of $\boldsymbol{y}^t$ and the second inequality is due to Assumption 1. Let us define $\eta_j := |\sum_{i\in N_j^t} p_i\hat{c}_{ij}^t - \sum_{i\in X} p_i\hat{c}_{ij}^t y_{ij}^*|$. If $\lambda_j^t < \beta_j$ (i.e., Part (i)), it holds that $u_j^* = 0$ and $\sum_{i\in N_j^t} p_i\hat{c}_{ij}^t y_{ij}^* = \sum_{i\in X} p_i\hat{c}_{ij}^t y_{ij}^* + u_j^* \geq \alpha$. Therefore, $\sum_{i\in N_j^t} p_i\hat{c}_{ij}^t \geq p_i\hat{c}_{ij}^t y_{ij}^* - \eta_j \geq \alpha - \eta_j$. The argument for the case $\lambda_j^t > 0$ (i.e., Part (ii)) follows similarly. □

B.3. Proof of Theorem 1

In this part, we present the main arguments for the proof of Theorem 1. Note that N_j^t and $\hat{c}_{ij}^t$ are random variables measurable with respect to the history $\mathcal{H}_t$. We define $\hat{\gamma}_j^t := \sum_{i\in N_j^t} p_i\hat{c}_{ij}^t$ and $\gamma_j^t := \sum_{i\in N_j^t} p_i c_{ij}^t$. One can also show that for the SBL algorithm, the expected number of any ad j being sampled before round t is $\sum_{s=1}^{t}\frac{\epsilon_s}{K} = \Theta\left(t^{\frac{2}{3}}K^{-\frac{2}{3}}(\log t)^{\frac{1}{3}}\right)$. By Hoeffding's inequality, by round t, ad j has been sampled $\Theta\left(t^{\frac{2}{3}}K^{-\frac{2}{3}}(\log t)^{\frac{1}{3}}\right)$ times with probability $1-t^{-4}$. This implies that, by Assumption 2 and the union bound, the CTR estimate satisfies $|\hat{c}_{ij}^t - c_{ij}^t| = O\left(t^{-\frac{1}{3}}(\log t)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right)$ for all arms with probability at least $1-t^{-3}$. Combining the above observations, we will show the following lemma.

LEMMA 3 **(Per Period Gap of the Alternative Reward Process)**. *Conditioned on exploitation at round t of the SBL algorithm, it holds that*

$$\mathbb{E}\left[\tau(x_t,a_t)\right] \geq \mathsf{OPT} + \mathbb{E}[\sum_{j=1}^{K}\beta_j(\gamma_j^t-\alpha)^+] - O\left(t^{-\frac{1}{3}}(\log t)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right) - O\left(T^{-\frac{1}{3}}(\log T)^{\frac{1}{3}}K^{\frac{1}{3}}\right). \tag{14}$$

Proof of Lemma 3.

We first apply the approximate complementary slackness (Lemma 2) to bound the expected empirical auxiliary reward process under the implementation of the dual-solution in the primal space:

$$\sum_{i\in N_j^t} p_i\hat{c}_{ij}^t(b_j+\beta_j).$$

Fixing the history $\mathcal{H}_t$ and an arbitrary ad j, we have the following equality:

$$\begin{aligned}\mathsf{OPT}^t &= \sum_{i\in N_j^t} p_i \max_{j'=1,2,\ldots,K}\left(\hat{c}_{ij'}^t(b_{j'}+\lambda_{j'}^t)\right) + \alpha(\beta_j-\lambda_j^t)\\ &= \sum_{i\in N_j^t} p_i\hat{c}_{ij}^t(b_j+\lambda_j^t) + \alpha(\beta_j-\lambda_j^t)\\ &= \sum_{i\in N_j^t} p_i\hat{c}_{ij}^t(b_j+\beta_j) - (\hat{\gamma}_j^t-\alpha)(\beta_j-\lambda_j^t),\end{aligned}$$

where the first equality follows from the definition of OPT^t, the second from the definition of N_j^t, and the third from the identity $\sum_{i\in N_j^t} p_i \hat{c}_{ij}^t = \hat{\gamma}_j^t$. Thus, we have

$$\sum_{i\in N_j^t} p_i \hat{c}_{ij}^t (b_j + \beta_j) = \mathsf{OPT}^t + (\hat{\gamma}_j^t - \alpha)(\beta_j - \lambda_j^t). \tag{15}$$

Hence, if the (exact) complimentary slackness condition holds with $\eta_j = 0$ for all $j \in A$ (see Lemma 2), we have $(\hat{\gamma}_j^t - \alpha)(\beta_j - \lambda_j^t) = \beta_j(\hat{\gamma}_j^t - \alpha)^+$. In this case,

$$\sum_{i\in N_j^t} p_i \hat{c}_{ij}^t (b_j + \beta_j) = \mathsf{OPT}^t + (\hat{\gamma}_j^t - \alpha)(\beta_j - \lambda_j^t) = \mathsf{OPT}^t + \beta_j(\hat{\gamma}_j^t - \alpha)^+$$

Otherwise, $\eta_j > 0$ for some $j \in A$ in Lemma 2. In this case, we show the following bound:

$$\sum_{j=1}^{K} (\hat{\gamma}_j^t - \alpha)(\beta_j - \lambda_j^t) \geq \sum_{j=1}^{K} \beta_j(\hat{\gamma}_j^t - \alpha)^+ - O\left(T^{-\frac{1}{3}} (\log T)^{\frac{1}{3}} K^{\frac{1}{3}}\right), \tag{16}$$

To obtain the inequality in (16), we observe, by Lemma 2, that $\sum_{j\in A} \eta_j \leq O(T^{-\frac{1}{3}} (\log T)^{\frac{1}{3}} K^{\frac{1}{3}})$, so it suffices to show that

$$(\hat{\gamma}_j^t - \alpha)(\beta_j - \lambda_j^t) \geq \beta_j(\hat{\gamma}_j^t - \alpha)^+ - \eta_j \text{ for all } j \in A.$$

More specifically, we consider three cases: (a) $\lambda_j^t = 0$, (b) $\lambda_j^t \in (0, \beta_j)$, and (c) $\lambda_j^t = \beta_j$.

If $\lambda_j^t = 0$, we have $(\hat{\gamma}_j^t - \alpha)(\beta_j - \lambda_j^t) = \beta_j(\hat{\gamma}_j^t - \alpha)$. If $\hat{\gamma}_j^t > \alpha$, clearly, $\beta_j(\hat{\gamma}_j^t - \alpha)^+ = \beta_j(\hat{\gamma}_j^t - \alpha)$. Otherwise, $(\hat{\gamma}_j^t - \alpha)^+ = 0$, and $\lambda_j^t = 0$ implies that $\eta_j \geq \alpha_j - \hat{\gamma}_j^t$. Therefore,

$$\beta_j(\hat{\gamma}_j^t - \alpha) = \beta_j(\hat{\gamma}_j^t - \alpha)^+ - \beta_j(\alpha - \hat{\gamma}_j^t) \geq \beta_j(\hat{\gamma}_j^t - \alpha)^+ - \beta_j\eta_j \geq \beta_j(\hat{\gamma}_j^t - \alpha)^+ - \eta_j,$$

where the last inequality follows from $\beta_j \in [0, 1]$.

If $\lambda_j^t = \beta_j$, we have $(\hat{\gamma}_j^t - \alpha)(\beta_j - \lambda_j^t) = 0$. Furthermore, the following inequality holds

$$0 = \beta_j(\hat{\gamma}_j^t - \alpha)^+ - \beta_j(\hat{\gamma}_j^t - \alpha)^+ \geq \beta_j(\hat{\gamma}_j^t - \alpha)^+ - \beta_j\eta_j \geq \beta_j(\hat{\gamma}_j^t - \alpha)^+ - \eta_j,$$

where the first inequality follows from $\hat{\gamma}_j^t - \alpha \leq \eta_j$ (see Lemma 2) and $\eta_j \geq 0$, which together imply $(\hat{\gamma}_j^t - \alpha)^+ \leq \eta_j$, and the second from $\beta_j \in [0, 1]$. It then follows that $(\hat{\gamma}_j^t - \alpha)(\beta_j - \lambda_j^t) \geq \beta_j(\hat{\gamma}_j^t - \alpha)^+ - \eta_j$ for the case where $\lambda_j^t = \beta_j$.

If $\lambda_j \in (0, \beta_j)$, Lemma 2 suggests that $|\hat{\gamma}_j^t - \alpha| \leq \eta_j$. In the case where $\hat{\gamma}_j^t > \alpha$, we have $\hat{\gamma}_j^t - \alpha \leq \eta_j$ and, therefore,

$$(\hat{\gamma}_j^t - \alpha)(\beta_j - \lambda_j^t) = (\hat{\gamma}_j^t - \alpha)^+\beta_j - (\hat{\gamma}_j^t - \alpha)\lambda_j^t \geq (\hat{\gamma}_j^t - \alpha)^+\beta_j - \eta_j\lambda_j^t \geq (\hat{\gamma}_j^t - \alpha)^+\beta_j - \eta_j,$$

where the first inequality follows from $\hat{\gamma}_j^t - \alpha \leq \eta_j$, and second from $\lambda_j^t < \beta_j \leq 1$. In the case where $\hat{\gamma}_j^t \leq \alpha$, we have

$$(\hat{\gamma}_j^t - \alpha)(\beta_j - \lambda_j^t) = (\hat{\gamma}_j^t - \alpha)^+\beta_j - (\alpha - \hat{\gamma}_j^t)(\beta_j - \lambda_j^t) \geq (\hat{\gamma}_j^t - \alpha)^+\beta_j - \eta_j\beta_j \geq (\hat{\gamma}_j^t - \alpha)^+\beta_j - \eta_j,$$

where the first equality follows from $(\hat{\gamma}_j^t - \alpha)^+ = 0$ in this case, the first inequality from $0 \leq \beta_j - \lambda_j^t \leq \beta_j$ and $0 \leq \alpha - \hat{\gamma}_j^t \leq \eta_j$, and the second inequality from $\beta_j \in [0, 1]$. Therefore, $(\hat{\gamma}_j^t - \alpha)(\beta_j - \lambda_j^t) \geq \beta_j(\hat{\gamma}_j^t - \alpha)^+ - \eta_j$ for all $j \in A$ and, hence, inequality (16) follows.

Finally, we evaluate $\mathbb{E}[\tau(x_t,a_t)|\mathcal{H}_t]$ and bound the terms OPT^t and $(\hat{\gamma}_j^t-\alpha)^+$. Consider two cases: (a) $|\hat{c}_{ij}^t-c_{ij}^t|=O\left(t^{-\frac{1}{3}}(\log t)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right)$ for all j, which occurs with probability at least $1-t^{-3}$ (see the discussions before Lemma 3); and (b) $|\hat{c}_{ij}^t-c_{ij}^t|\neq O\left(t^{-\frac{1}{3}}(\log t)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right)$ for some $j\in A$, which occurs with probability less than t^{-3}.

We first consider the case where $|\hat{c}_{ij}^t-c_{ij}^t|=O\left(t^{-\frac{1}{3}}(\log t)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right)$ for all ad j (which occurs with probability at least $1-t^{-3}$). It follows from the definition of OPT (see Lemma 1) that

$$\mathsf{OPT}=\min_{0\le\lambda_j\le\beta_j}\sum_{i\in X}p_i\max_{j=1,2,\dots,K}\left(c_{ij}(b_j+\lambda_j)\right)+\alpha\sum_{j=1}^{K}(\beta_j-\lambda_j).$$

Because $|\hat{c}_{ij}^t-c_{ij}^t|=O\left(t^{-\frac{1}{3}}(\log t)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right)$, by the definitions of OPT and OPT^t, we have

$$\mathsf{OPT}^t\ge\mathsf{OPT}-O\left(t^{-\frac{1}{3}}(\log t)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right). \tag{17}$$

Similarly, we bound $\mathbb{E}[\tau(x_t,a_t)|\mathcal{H}_t]$ by

$$\mathbb{E}[\tau(x_t,a_t)|\mathcal{H}_t]=\sum_{j=1}^{K}\sum_{i\in N_j^t}p_ic_{ij}^t(b_j+\beta_j)\ge\sum_{j=1}^{K}\sum_{i\in N_j^t}p_i\hat{c}_{ij}^t(b_j+\beta_j)-O\left(t^{-\frac{1}{3}}(\log t)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right). \tag{18}$$

Furthermore, by Jensen's inequality, we observe that

$$(\hat{\gamma}_j^t-\alpha)^+\ge\left(\gamma_j^t-\alpha-O\left(t^{-\frac{1}{3}}(\log t)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right)\sum_{i\in N(j)}p_i\right)^+\ge(\gamma_j^t-\alpha)^+-O\left(t^{-\frac{1}{3}}(\log t)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right). \tag{19}$$

Collecting the terms of (15), (16), (17), (18), and (19) above, we have that

$$\mathbb{E}\left[\tau(x_t,a_t)\middle|\mathcal{H}_t\right]\ge\mathsf{OPT}+\sum_{j=1}^{K}\beta_j(\gamma_j^t-\alpha)^+-O\left(t^{-\frac{1}{3}}(\log t)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right)-O\left(T^{-\frac{1}{3}}(\log T)^{\frac{1}{3}}K^{\frac{1}{3}}\right).$$

For the case $|\hat{c}_{ij}^t-c_{ij}^t|\neq O\left(t^{-\frac{1}{3}}(\log t)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right)$, which occurs with probability less than t^{-3}, we can bound the expected gap between $\mathbb{E}[\tau(x_t,a_t)]$ and OPT by $O(t^{-3})$, which is a lower order term compared to the gap in the case where $|\hat{c}_{ij}^t-c_{ij}^t|=O\left(t^{-\frac{1}{3}}(\log t)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right)$. Integrating over the distribution of history $\mathcal{H}_t$, we have established inequality (14). □

Summing inequality (14) over the whole T periods (i.e., $t=1,2,...,T$) and invoking Jensen's Inequality, we have the following regret bound on the expected auxiliary reward process $\mathbb{E}\left[\sum_{t=1}^{T}\tau(x_t,a_t)\right]$ under the SBL algorithm:

$$\mathbb{E}\left[\sum_{t=1}^{T}\tau(x_t,a_t)\right]\ge T\cdot\mathsf{OPT}+\mathbb{E}\left[\sum_{j=1}^{K}\beta_j\left(\sum_{t=1}^{T}\gamma_j^t-\alpha T\right)^+\right]-O\left(T^{\frac{2}{3}}(\log T)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right), \tag{20}$$

Next, we bound the difference between the auxiliary reward process $\tau(x_t,a_t)$ and the true reward process $r(x_t,a_t)$. Denote the total number of clicks for ad j in all T rounds as $n(j)$.

LEMMA 4 **(Difference between Two Reward Processes)**. *Under the SBL algorithm, it holds that*

$$\mathbb{E}\left[\sum_{j=1}^{K}\left(n(j)-\sum_{t=1}^{T}\gamma_j^t\right)^+\right]\le O\left(\sqrt{KT\log T}\right). \tag{21}$$

Proof of Lemma 4.

It suffices to establish a high probability bound: With probability at least $1-T^{-4}$, the following inequality holds:

$$\sum_{j=1}^{K}\left(n(j)-\sum_{t=1}^{T}\gamma_j^t\right)^+ \le O\left(\sqrt{KT\log T}\right). \tag{22}$$

We now show that for any subset of ads, denoted by $\mathcal{S}$,

$$\sum_{j\in\mathcal{S}} n(j)-\sum_{j\in\mathcal{S}}\sum_{t=1}^{T}\gamma_j^t \le O\left(\sqrt{KT\log T}\right) \text{ with probability at least } 1-T^{-4}. \tag{23}$$

Note that given the history by round t, $\mathcal{H}_t$, the expected total number of clicks for ad j is given by γ_j^t. One can use Azuma-Hoeffding inequality to show that for a fixed subset $\mathcal{S}$, we have with probability at most T^{-4K},

$$\sum_{j\in\mathcal{S}} n(j)-\sum_{j\in\mathcal{S}}\sum_{t=1}^{T}\gamma_j^t \ge O\left(\sqrt{KT\log T}\right).$$

Take a union bound over all subsets and notice that $2^K T^{-4K}\le T^{-4}$. Hence, (23) holds with probability at least $1-T^{-4}$.

We now show (22) by contradiction. Suppose that (22) does not hold with probability at least T^{-4}. Define $\mathcal{S}'$ as the set of ads such that $n(j)>\sum_{t=1}^{T}\gamma_j^t$. It follows that

$$\sum_{j\in\mathcal{S}'} n(j)-\sum_{j\in\mathcal{S}'}\sum_{t=1}^{T}\gamma_j^t=\sum_{j\in\mathcal{S}'}\left(n(j)-\sum_{t=1}^{T}\gamma_j^t\right)=\sum_{j=1}^{K}\left(n(j)-\sum_{t=1}^{T}\gamma_j^t\right)^+ > O\left(\sqrt{KT\log T}\right),$$

with probability at least T^{-4}, which contradicts inequality (23). Thus, inequality (22) holds with probability at least $1-T^{-4}$. Finally, we take the expectation of (22) and the inequality (21) follows immediately. □

We are now ready to prove Theorem 1.

Proof of Theorem 1.

First, we have the following inequality:

$$\begin{aligned}\sum_{t=1}^{T}\tau(x_t,a_t)&=\sum_{t=1}^{T}r(x_t,a_t)+\sum_{j=1}^{K}\beta_j\left(n(j)-\alpha T\right)^+\\&\le\sum_{t=1}^{T}r(x_t,a_t)+\sum_{j=1}^{K}\beta_j\left(n(j)-\sum_{t=1}^{T}\gamma_j^t\right)^+ +\sum_{j=1}^{K}\beta_j\left(\sum_{t=1}^{T}\gamma_j^t-\alpha T\right)^+,\end{aligned} \tag{24}$$

where the inequality follows from $(X+Y)^+\le X^++Y^+$ for any $X,Y\in\mathbb{R}$. Putting the inequalities (20), (21), and (24) together, we obtain, for the exploitation rounds of the SBL algorithm,

$$\begin{aligned}\mathbb{E}\left[\sum_{t=1}^{T}r(x_t,a_t)\right]\ge&\mathbb{E}\left[\sum_{t=1}^{T}\tau(x_t,a_t)\right]-\mathbb{E}\left[\sum_{j=1}^{K}\beta_j\left(n(j)-\sum_{t=1}^{T}\gamma_j^t\right)^+\right]-\mathbb{E}\left[\sum_{j=1}^{K}\beta_j\left(\sum_{t=1}^{T}\gamma_j^t-\alpha T\right)^+\right]\\\ge&\mathbb{E}\left[\sum_{t=1}^{T}\tau(x_t,a_t)\right]-O\left(\sqrt{KT\log T}\right)-\mathbb{E}\left[\sum_{j=1}^{K}\beta_j\left(\sum_{t=1}^{T}\gamma_j^t-\alpha T\right)^+\right]\\\ge&T\cdot\mathsf{OPT}-O\left(\sqrt{KT\log T}\right)-O\left(T^{\frac{2}{3}}(\log T)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right)\\=&T\cdot\mathsf{OPT}-O\left(T^{\frac{2}{3}}(\log T)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right).\end{aligned}$$

Furthermore, we notice that the expected number of exploration rounds is of order $O\left(T^{\frac{2}{3}}(\log T)^{\frac{1}{3}}K^{\frac{1}{3}}\right)$, which implies that the regret induced by exploration in SBL is also bounded by $O\left(T^{\frac{2}{3}}(\log T)^{\frac{1}{3}}K^{\frac{1}{3}}\right)$. Therefore, with the additional assumptions that (a) $\tau_m - \tau_{m-1} = 1$ for each m and (b) p_i is known to the algorithm for each i, the expected regret is bounded by an order of $O\left(T^{\frac{2}{3}}(\log T)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right)$.

Finally, we relax the above two assumptions (i.e., $\tau_m - \tau_{m-1} = 1$ for each m and p_i is known to the algorithm for each i) by demonstrating that imposing them will only incur an additional regret of an order lower than $O\left(T^{\frac{2}{3}}(\log T)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right)$. To begin with, we show that it suffices to solve the empirical dual problem with a fixed epoch of size $O(T^{\frac{2}{3}})$. Since the regret bound is of order $\tilde{O}(T^{\frac{2}{3}})$, we can discard the first $T^{\frac{2}{3}}$ periods without affecting the order of the regret bound. After the first $T^{\frac{2}{3}}$ periods, with a fixed epoch schedule such that $\tau_{m+1} - \tau_m = O(T^{\frac{2}{3}})$, we have $\tau_m \geq (1/2)\tau_{m+1}$. Therefore, at round t the additional regret it incurs to solve the empirical dual program is at most a constant multiplication of $O\left(t^{-\frac{1}{3}}(\log t)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right)$, which is still of order $O\left(t^{-\frac{1}{3}}(\log t)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right)$. Therefore, summing this bound over t from 1 to T, we have that the total additional regret is of order $O\left(T^{\frac{2}{3}}(\log T)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right)$.

Next, we show that using the empirical probability $\hat{p}_i$ (instead of the true one p_i) only incurs an additional regret of an order lower than the one in Lemma 3, i.e., $O\left(t^{-\frac{1}{3}}(\log t)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right) + O\left(T^{-\frac{1}{3}}(\log T)^{\frac{1}{3}}K^{\frac{1}{3}}\right)$. Note that, by Assumption 2, the estimate satisfies $|\hat{c}_{ij}^t - c_{ij}^t| = O\left(t^{-\frac{1}{3}}(\log t)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right)$ for all ads with probability at least $1 - t^{-3}$. Furthermore, the empirical distribution estimate $\hat{p}_i^t$ satisfies that for any context $i \in X$, with probability at least $1 - t^{-4}$, we have $|\hat{p}_i^t - p_i| \leq O(\sqrt{\log t/t})$ (see the discussions in Appendix B.2). Combining the above two error estimation bounds on $\hat{c}_{ij}^t$ and $\hat{p}_j$, we have, by the definitions of OPT^t and OPT,

$$\mathsf{OPT}^t \geq \mathsf{OPT} - O\left(t^{-\frac{1}{3}}(\log t)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right) - O\left(t^{-\frac{1}{2}}(\log t)^{\frac{1}{2}}\right) - O\left(t^{-\frac{5}{6}}(\log t)^{\frac{5}{6}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right) = \mathsf{OPT} - O\left(t^{-\frac{1}{3}}(\log t)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right).$$

Hence, Lemma 3 and inequality (14) continue to hold if we replace p_i with $\hat{p}_i^t$ for each context i and each round t. The rest of Theorem 1's proof remains the same. Summarizing our argument above, we have shown that the expected regret of the SBL algorithm is bounded by $O\left(T^{\frac{2}{3}}(\log T)^{\frac{1}{3}}K^{\frac{1}{3}}d^{\frac{1}{2}}\right)$. □

Appendix C: Details of the PID system

As introduced in Section 3, the PID controller is a feedback control device widely used in online advertising platforms, especially for the oCPC and oCPM billing options. The PID controller aims to gear the realized CPA of each ad as close to the target CPA as possible, so it increases the bid price thus boosting its eCPM and the chance of winning the auction, if the actual cost per conversion of an ad falls below the target cost, and vice versa. Formally, the PID controller is formulated as follows:

$$\begin{aligned}
&\mathsf{error}_t = \mathsf{targetBid} - \mathsf{realCost}_t/\mathsf{realConversion}_t,\\
&\mathsf{P}_t = k_p \times \mathsf{error}_t\\
&\mathsf{I}_t = k_i \times \sum_{t' \leq t} \mathsf{error}_{t'},\\
&\mathsf{D}_t = k_d \times (\mathsf{error}_t - \mathsf{error}_{t-1})\\
&\mathsf{PID}_t = \mathsf{P}_t + \mathsf{I}_t + \mathsf{D}_t,\\
&\mathsf{systemBid}_{t+1} = \mathsf{systemBid}_t + \mathsf{PID}_t \times \mathsf{systemBid}_t
\end{aligned}$$

It is clear from the above formulation that the PID controller changes the system bidding price $\mathsf{systemBid}_t$ after accumulating the feedback data for each ad within a fixed amount of time. The first equation quantifies the gap between the target cost and the real cost (the total actual cost divided by the total actual conversions). P_t, I_t, D_t represents the Proportional, Integral, Derivative (PID) term in the PID system respectively. And the corresponding coefficients k_p, k_i, k_d are hyper-parameters to be fine-tuned. Readers interested in more details about the PID system are referred to, e.g., Yang et al. (2019) and Zhang et al. (2016).

Appendix D: Numerical Experiments in the Simulation System

In this section, we present our numerical experiments in more detail. We first discuss the data set and the simulation model, with which we conduct two numerical experiments. The first experiment is discussed in Section 4.3, which shows that our SBL algorithm indeed incurs a sublinear reward loss. The second numerical experiment, as we will show in Section D.1, quantifies the long-term monetary value of the algorithm and, therefore, further demonstrates that the SBL algorithm can successfully trade off the short-term revenue and the long-term benefit of cold start. The third numerical experiment in Section D.2 quantifies the estimation biases under single-sided experiments (see also Section 5). Finally, in Section D.3, we use simulation to show how we could extrapolate the estimates from two-sided experiments to the entire population.

Data Set. As mentioned in Section 4.3, our data set contains one-week log records of 300 ads, which consists of three types of data from December 1, 2019 to December 7, 2019, including (1) impression, click and conversion records of the ads, (2) the user page-view records with gender, location, age, device features of each user displayed with the ads, and (3) the bidding prices of each ad. However, to protect the platform's sensitive data, all the reported values in this section are re-scaled.

Advertising Mechanism. To simplify the advertising mechanism of our simulator, we only consider the CPC billing option and first-price auction in our simulation. Thus, we ignore the conversions and focus on clicks. We generate the incoming users' ad view requests with features using the shuffled real-world user page-view records. Each ad joins the auction with the true bidding price and the pCTR generated by a logistic regression model described below. After allocating an ad to the user page view, the simulator gets the feedback whether the ad is clicked based on the ground truth CTR and a binomial click probability. If the ad is clicked, the advertiser is charged the bidding price for each click. Meanwhile, the pCTR model is updated with this new feedback data of user click.

Ground Truth CTR Model and pCTR Model. The simulator is equipped with a ground truth click-through model for each ad via fitting a logistic regression model on its historical impression and click records. The pCTR model is essential in deciding which ad will win each user impression. In our simulator, the pCTR model is built upon a logistic regression model with randomly initialized parameters, which are updated in real-time based on the click-through feedback.

D.1. Short-term and Long-term Value Analysis

We use our simulator to numerically demonstrate that our SBL algorithm can successfully trade off the short-term revenue and the long-term value of cold start.

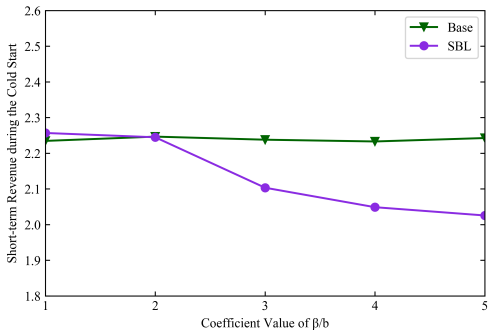
To mimic the real-world advertising platform where mature ads and new ads are mixed together, we randomly classify the 300 ads in the simulation into 100 new ads and 200 mature ads. The pCTR models

of mature ads are pre-trained before the start of simulation. The simulation contains two phases, the cold start phase and the long-term stationary phase. During the cold-start phase, there are in total 1,000,000 user page-views to be allocated. After the cold start phase, consistent with Figure 1, we assume that a new ad with fewer accumulated clicks have a higher probability to leave the advertising platform. The new ads who stay on the platform together with the 200 mature ads proceed to the stationary phase with another 1,000,000 user page-views. To evaluate our algorithm, we compare the current cold start practice of Platform O (the PID-based strategy) to our SBL algorithm with different $\boldsymbol{\beta}$ values in this two-phase simulation. Note that both our new algorithm and the current cold start practice of Platform O are only implemented during the cold start phase. Following the current practice, we homogeneously increase the bidding prices of new ads for the first 1,000,000 page views as the baseline.

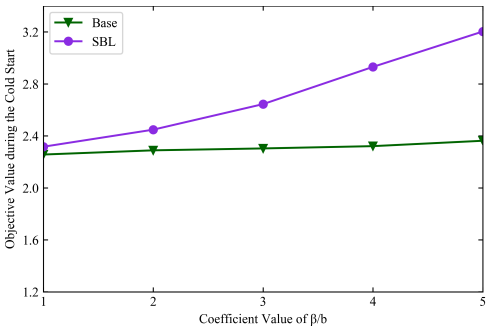
Recall that we separate 2,000,000 page views equally into the cold start phase and the stationary phase. We capture ads' retention behavior after the cold start phase (see Figure 1), using a piecewise linear model to fit the real data with the following specification:

$$\mathbb{P}[\mathsf{retention}] = \gamma_1 + \gamma_2 \times \min\left\{1, \#\mathsf{clicks}/\alpha T\right\},$$

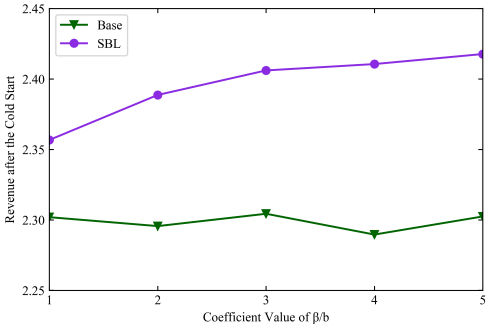
where $\#\mathsf{clicks}$ is the number of accumulated clicks during the cold start phase, $\alpha T = 1000$ is the target number of clicks with $\alpha = 0.001$ and $T = 1,000,000$. To protect the sensitive data, we are required not to report the coefficients γ_1 and γ_2.



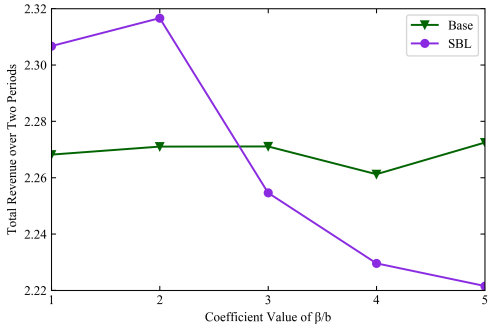
(a) Short-term Empirical CPM



(b) Short-term Objective Value Per Mille



(c) Long-term Empirical CPM



(d) Average Empirical CPM over Two Phases

Figure 8 Simulation Results under the oCPM and Second-price Auction Setting

We run the simulation with different values of $\boldsymbol{\beta} \in \{\boldsymbol{b}, 2\boldsymbol{b}, 3\boldsymbol{b}, 4\boldsymbol{b}, 5\boldsymbol{b}\}$, each with 5 replications and report the average value in Figure 8. The line labeled "SBL" represents the performance of our new algorithm, while the line labeled "base" is the baseline strategy following the current cold start practice on Platform O. The reported monetary revenue and the objective value are scaled as the value per thousand page views, namely the empirical cost per mille (eCPM). As $\boldsymbol{\beta}$ increases, the weight of new ads increases in our algorithm in the cold start phase, resulting in a decreased short-term revenue (Figure 8(a)), and an increased total objective value (Figure 8(b)). As for the performance of long-term revenue, we observe from 8(c) that new ads with larger $\boldsymbol{\beta}$ values gain more impressions and clicks during the cold start phase. Thus, they have a higher probability to stay on the platform, which enlarges the pool of ads and increases the long-term revenue during the stationary phase. In Figure 8(d), we illustrate how different choices of $\boldsymbol{\beta}$ will impact the total value of the SBL algorithm, measured by the average empirical CPM over the short-term cold start phase and the long-term stationary phase. As shown in Figure 8(d), overly high values of $\boldsymbol{\beta}$ (e.g., $\boldsymbol{\beta} \in \{3\boldsymbol{b}, 4\boldsymbol{b}, 5\boldsymbol{b}\}$) yield substantial revenue losses in the cold start phase, which cannot be compensated by the long-term revenue boost in the stationary phase. This result suggests that, in the real-world implementation, we should carefully choose the cold start parameter values based on rigorous empirical estimations of the long-term value of new ads which are successfully cold started.

D.2. Estimation Biases with One-sided Experiments

To illustrate the potential estimation biases induced by the violation of SUTVA with Ad-side and UV-side experiments, we run three numerical simulations (Ad-side randomization experiment (see Figure 5(a), UV-side randomization experiment (see Figure 5(b), and two-sided experiment (see Figure 6) in our simulation system with 100 new ads and 200 mature ads, as well as 1,000,000 user views. We fix $\alpha = 0.001$ and $\boldsymbol{\beta} = 2\boldsymbol{b}$ for all experiments in this set of simulations. In all these simulations, 50% of UVs/Ads are randomly assigned into the treatment condition, and the other 50% into the control condition.

To quantify the true treatment effect of the SBL algorithm applied to the full population, we run another two simulations, one with the benchmark algorithm applied to all ads and full user traffic, the other with the SBL algorithm applied to all new ads and all users. The comparison between this two simulations shows that the SBL algorithm boosts the cold start success rate from 4% to 6%. So we use the 2% increase in the cold start success rate as the ground truth average treatment effect of the SBL algorithm applied to the entire population of ads and users.

We replicate the simulation for each randomized experiment for five times and report corresponding estimation results of cold start success rate in Table 6. Our simulation results demonstrate that the ad-side experiment significantly overestimates the treatment effect of the proposed algorithm, whereas the user-side experiment underestimates the effect. Furthermore, the two-sided equips us with an unbiased estimate. Therefore, our simulation results necessitate and validate our two-sided experiment design by showing that whereas one-sided experiments are likely to produce substantially biased estimates, our novel two-sided design helps correct such biases.

Table 6 Bias Analysis of Different Estimators

	Ad-Side Experiment		UV-Side Experiment		Two-Side Experiment	
	Treatment	Control	Treatment	Control	Treatment	Control
Cold-Start Success Rate	0.068	0.024	0.050	0.038	0.060	0.038
	(0.011)	(0.017)	0.007	0.013	0.007	0.008
Value of Estimator	4.4%**		1.2%*		2.2%**	
Bias/Global Treatment Effect	120%		-40%		10%	

Note: *p<0.05; **p<0.01; ***p<0.001; ****p<0.0001. Standard errors are reported in the parentheses.

D.3. Extrapolation of Two-sided Experiments to the Entire Population

We now use simulation to examine how to extrapolate the result of a two-sided experiment to estimate the effect of an algorithm applied to the entire population. With the same setup for two-sided experiments in Appendix D.2, we run five simulations with different ad and user traffics assigned into the treatment and control groups. In each simulation, the same proportion (10%, 30%, 50%, 70%, and 90%) of ads and users are assigned into the treatment condition. For each scenario, we replicate the simulation for ten times. Figure 9 report our simulation results with different ad and user traffics. As shown in Figure 9, the number of new ads which successfully cold start in the treatment (resp. control) condition increases (resp. decreases) almost linearly in the treatment traffic. As such, the estimated average treatment effect of the new algorithm on the cold start success rate remains to be around 2% (2.33%, 1.90%, 1.40%, 2.38%, and 2.22% for 10%, 30%, 50%, 70%, and 90% treatment traffic, respectively). Therefore, for our simulation system, we could linearly extrapolate the estimate from two-sided experiments to the entire population in an unbiased fashion. Consistent with the standard statistics theory, our simulation also shows that the variance of the estimate is larger with a more imbalanced treatment-control traffic.

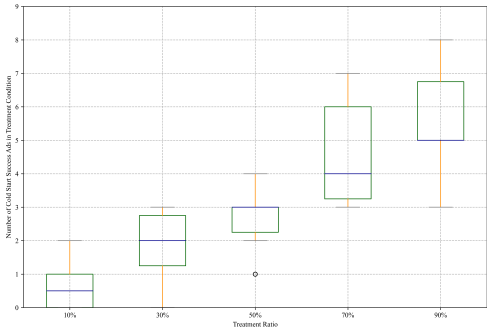
Appendix E: Subgradient Descent Algorithm

To demonstrate the effectiveness of our subgradient descent algorithm to obtain the shadow bids $\boldsymbol{\lambda}$, we compare with the objective values from (a) our approach, (b) the Simplex method which solves the primal directly, (c) another gradient-based method SHALE (Bharadwaj et al. 2012, Hojjat et al. 2017), and (d) the current practice of Platform O, namely showing the ad with maximum eCPM without considering the cold start value. We examine a small-scale instance with 100 Ads and 10,000 UVs. However, our online implementation solves the dual instances with more than 10,000,000 UVs, which is impossible to solve in a reasonable time by the standard Simplex approach. The stopping condition of our subgradient descent algorithm is when the duality gap is less than $O(10^{-4})$. Other parameters such as bidding prices b_j and pCTRs are directly from the real data. The computational results are summarized in Table 7.

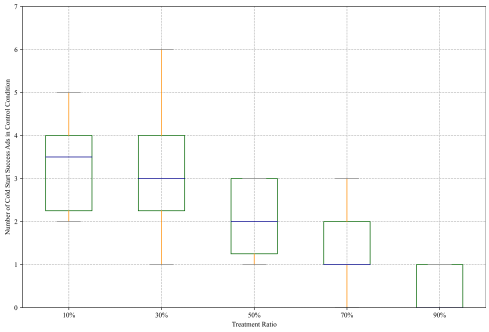
Table 7 Objetive Value Comparison

	Current Practice (1)	SHALE (2)	Subgradient Descent (3)	Simplex (4)
Revenue	288,556	284,913	278,598	278,588
Cold Start Reward	67,747	76,171	98,429	98,679
Objective Value	356,303	361,084	377,026	377,267

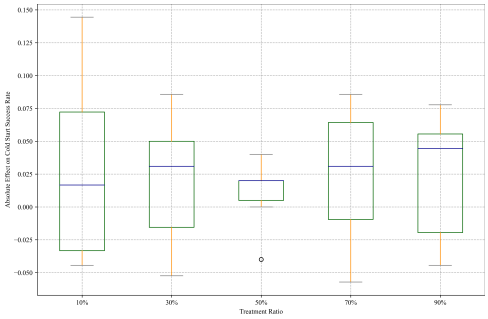
The objective value is the sum of the revenue and cold start reward. The relative difference between our dual-based subgradient descent approach and the optimal objective value is less than 0.07%, which



(a) Number of Success New Ads in Treatment Condition



(b) Number of Success New Ads in Control Condition



(c) Absolute Effect on Cold Start Success Rate

Figure 9 Simulation Results with Extrapolation

suggests that the gap induced by integer round-offs and the stopping condition in our algorithm is negligible. Moreover, our algorithm performs substantially better than the SHALE algorithm (Bharadwaj et al. 2012).

Appendix F: Robustness Check of the UV Sampling Rate

In this section, we conduct robustness check for the UV sampling rate, which shows that even a low sampling rate of 1% for user views could already cover most of the new ads and produce robust dual solutions. Moreover, considering that both memory and computational time increase linearly with the sampling rate, we choose 4% sampling rate for our online implementation, which strikes a good balance of sample representativeness and computational time.

Table 8 Robustness Check of the UV Sampling Rate

	Sampling Rate of UV r		
	$r = 0.04$	$r = 0.02$	$r = 0.01$
	(1)	(2)	(3)
The number of new Ads	6216	6216	6216
Mean of λ	64.21	64.22	64.25
Standard deviation of λ	62.96	62.96	63.05
25^{th} *percentile of* λ	15.00	15.00	15.00
50^{th} *percentile of* λ	57.60	57.60	57.21
75^{th} *percentile of* λ	90.00	90.00	90.00

Note: The differences between λ's calculated by different sampling rates are not significant. P-values of t-tests between (1) and (2), (1) and (3), and (2) and (3) are, respectively, 0.716, 0.155, and 0.280.

Appendix G: Training with Neural Networks

In this section, we show that the fully connected neural networks satisfy *Prediction Oracle* with high probability (i.e., Assumption 2 holds) for both (a) the lazy training regime and (b) training with the gradient descent algorithm.

Before presenting the proof, we first introduce the fully connected neural network and the initialization procedure. A large-scale DSP like Platform O is typically armed with neural networks to predict the CTR and CVR of the ads thereof. Specifically, Platform O uses a set of fully connected neural networks with the ReLU as the activation function, i.e., $\sigma(x) = \max\{x, 0\}$. Without loss of generality, we assume all hidden layers of the neural network have the same number of nodes. And we denote L as the network depth, w as the number of nodes in each hidden layer, and w_0 as the dimension of the context feature space, i.e., $x_i \in \mathbb{R}^{w_0}$ for all $i \leq m$. Following the convention of the neural network literature (e.g., Cao and Gu 2019, Chizat et al. 2019), we parameterize the neural network by $\theta \in \mathbb{R}^d$, where the effective dimension $d = O(w^2 L + w w_0)$. Then, we can use the function $H_j(x_i, \theta) = \sqrt{w}\boldsymbol{W}_L \sigma(\boldsymbol{W}_{L-1}\sigma(\cdots \sigma(\boldsymbol{W}_1 x_i)))$ to represent the output of the neural network given the parameter θ, for any ad j and context i, where $\theta = [\text{vec}(\boldsymbol{W}_1), \cdots, \text{vec}(\boldsymbol{W}_L)]$, where the operator $\text{vec}(\cdot)$ refers to representing the matrix as a vector.

In practice, the initialization procedure may take the domain knowledge and from experience. In our analysis that follows, we adopt the initialization procedure in He et al. (2015), known as the *He Initialization* to set θ_0. For each layer $1 \leq l \leq L-1$, we set $\boldsymbol{W}_l$ to be $\left(\begin{smallmatrix} \boldsymbol{W} & 0 \\ 0 & \boldsymbol{W} \end{smallmatrix}\right)$ where each entry of this matrix $\boldsymbol{W}$ is randomly and independently drawn from a normal distribution $N(0, 2/w)$. The parameter of the last layer is initialized as $(\boldsymbol{w}^T, -\boldsymbol{w}^T)$ where each entry of vector $\boldsymbol{w}$ is randomly and independently drawn from distribution $N(0, 1/w)$. One can verify that under this initialization procedure of θ_0, it holds that $H_j(x_i, \theta_0) = 0$ for all i and j (see Cao and Gu 2019, He et al. 2015).

To validate Assumption 2 for fully connected neural networks, we make an additional technical assumption as follows, which is mild and commonly made in the related literature (e.g., Cao and Gu 2019, Zhou et al. 2020).

Assumption 3 **(Finite and Non-parallel Contexts)**. *(a) At each round $t \in [T]$, the context x_t is i.i.d. drawn from a finite context set $X = \{x_i\}_{i=1}^m$, where and $m = |X| \leq O(T^{1/4})$. (b) For any pair of contexts $x_i, x_j \in X$ $(i \neq j)$, x_i and x_j are not parallel. (c) The L_2-norm of each context is normalized to 1, i.e., $\|x_i\|_2 = 1$ for any context $x_i \in X$.*

G.1. Lazy Training Regime

The recent progress of *Neural Tangent Kernels* (e.g., Cao and Gu 2019, Jacot et al. 2018, Arora et al. 2019, Zhou et al. 2020) theoretically characterizes the representation power of a neural network. Following this literature, we use $\boldsymbol{H}$ to denote the Neural Tangent Kernel Matrix in the same way as Definition 4.1 of Zhou et al. (2020). Similar to Zhou et al. (2020) Assumption 4.2, we further assume that $\boldsymbol{H} \succeq \gamma I$ always holds for some $\gamma > 0$, where I is the identity matrix. This ensures that the Neural Tangent Kernel Matrix $\boldsymbol{H}$ is always non-singular. Lemma 5 below shows that, as long as the ground truth CTR can be represented as a bounded function of user contexts and ads, then the fully connected neural network with a large width w can accurately predict this CTR with high probability in terms of the *He Initialization*.

LEMMA 5 **(Lemma 5.1 in Zhou et al. (2020))**. *Under Assumption 3, there exists a constant $C > 0$ such that for any $\delta \in (0,1)$, if $w \geq Cm^4L^6\log(m^2L/\delta)/\gamma^4$, then with probability $1-\delta$ over the He Initialization of the parameter θ_0, there exists $\theta^* \in \mathbb{R}^d$ such that, for any $x_i \in X$, $j \in [K]$,*

$$c_{ij} = \langle \nabla_\theta H_j(x_i, \theta_0), \theta^* - \theta_0 \rangle, \sqrt{w}\,\|\theta^* - \theta_0\|_2 \leq \sqrt{2\boldsymbol{c}^T \boldsymbol{H} \boldsymbol{c}},$$

where $\boldsymbol{c} := [c_{ij}] \in \mathbb{R}^m$.

Notice that the approximation of ground truth CTR in Lemma 5 is linear in the gradient $\nabla_\theta H_j(x_i, \theta_0)$ parametrized by $\theta^* - \theta_0$. The original neural network mapping $H_j(\cdot,\cdot)$ is now divided into two steps. First, it maps the context x_i and the ad j to the gradient $\nabla_\theta H_j(x_i, \theta_0)$. This is a static mapping that depends on the initialization θ_0, but independent of the parameter θ. The second step linearly maps the gradient $\nabla_\theta H_j(x_i, \theta_0)$ to the true CTR, c_{ij}. As a consequence, to train the neural network under this regime, it suffices to fit a linear function parameterized by θ. This training method is referred to as *Lazy Training* in the literature (Chizat et al. 2019). Specifically, lazy training with the regularized least square loss function is equivalent to solving the following minimization problem:

$$\min_\theta w\lambda_0 \|\theta - \theta_0\|^2 + \sum_{s \in \mathcal{H}_t^j} (z_s - \langle \nabla_\theta H_j(x_s, \theta_0), \theta - \theta_0 \rangle)^2, \tag{25}$$

where λ_0 is the regularized parameter, z_s denotes the click-through outcome of round s and $\mathcal{H}_t^j$ denotes the time periods (until round t) in which ad j is displayed. With lazy training, we effectively linearize the neural network for CTR prediction, thus reducing it to a linear regression model. Lazy training facilitates us to focus on cold start algorithm design, without delving into the details of how a neural network shall be trained. Similar approaches, usually referred to as *Kernelized Contextual Bandits*, have been adopted in the contextual bandit literature (e.g., Valko et al. 2013). As identified by Chizat et al. (2019), the lazy training phenomenon, where a neural network behaves similarly to a linear model when the parameter θ is close to the initialization parameter θ_0, will occur when the neural network is over-parameterized. In addition, Chizat et al. (2019) also show that the gradient flows of the lazy training process and the gradient descent training process (see Appendix G.2) are closed for over-parameterized neural networks. We also remark that the real online training procedure of Platform O's CTR/CVR prediction model is neither pure supervised learning nor lazy training, but a substantial compromise under limited computational resources. In this regard, incorporating the exact online training process into our regret analysis is unnecessary and beyond the scope of this paper. Although Chizat et al. (2019) empirically shows that lazy training might not perform well in some cases with biased gradients, this training method still provides a good theoretical understanding of how $\hat{c}_{ij}$ is estimated under neutral networks, and inspires us to validate Assumption 2 for neural networks with gradient descent training (see Appendix G.2). We are now ready to validate Assumption 2 for neural networks under the lazy training regime.

PROPOSITION 1 **(Prediction Oracle with Lazy Training)**. *Under Assumption 3, the predicted CTR in each round t, $\hat{c}_{ij}^t$, is obtained by the lazy training procedure (25) of the neural network with $0 < \lambda_0 \leq O(\sqrt{1/(2w\boldsymbol{c}^T\boldsymbol{H}\boldsymbol{c})})$. We have there exists a constant $C > 0$, such that for any $\delta \in (0,1)$, if $w \geq$*

$Cm^4L^6\log(m^2L/\delta)/\gamma^4$, with probability $1-\delta$, it holds that, if ad j is displayed for $n_j^t \geq \Omega(d\log T)$ times independently before round t and the random click-through outcomes $\{v_s(a_s)\in\{0,1\}: 1\leq s\leq t-1\}$ are observed, then with probability at least $1-T^{-4}$, the following inequality holds for any context $x_i\in X$:

$$|\hat{c}_{ij}^t - c_{ij}| \leq O\left(\sqrt{\frac{d\log T}{n_j^t}}\right).$$

Proof of Proposition 1.

We first introduce some definitions. Let $\mathcal{H}_j^t$ be the set of user views where ad j is displayed before time t. We use I_d to denote the identity matrix with dimension d. Note that the cardinality of $|\mathcal{H}_j^t| = n_j^t$ by definition. We define $g_{ij} := \nabla_\theta H_j(x_i,\theta_0)$,for all $1\leq i\leq m$ and $j\in[K]$. Following the standard lazy training with regularized squared loss (25), we can compute θ^t in closed form at each round t as follows:

$$\begin{aligned} A_j^t &:= w\lambda_0 I_d + \sum_{i'\in\mathcal{H}_j^t} g_{i'j}g_{i'j}^T, \quad & D_j^t &:= [g_{i'j}^T]_{i'\in\mathcal{H}_j^t} \\ b_j^t &:= \sum_{i'\in\mathcal{H}_j^t} v_{i'j}g_{i'j}, & V_j^t &:= [v_{i'j}]_{i'\in\mathcal{H}_j^t} \\ \theta^t &:= (A_j^t)^{-1}b_j^t + \theta_0, & s_{ij}^t &:= \sqrt{g_{ij}^T(A_j^t)^{-1}g_{ij}} \end{aligned}$$

By Lemma 5, we consider the case, with high probability $1-\delta$, where CTR c_{ij} can be perfectly predicted via a linear mapping. Thus, at round t after observing n_j^t samples of each ad j and conditioned on $\mathcal{H}_j^t$, we have for a fixed context i, with probability at least $1-\delta$:

$$\begin{aligned} |\hat{c}_{ij}^t - c_{ij}| &= |g_{ij}^T(\theta^t-\theta_0) - g_{ij}^T(\theta^*-\theta_0)| \\ &= |g_{ij}^T(A_j^t)^{-1}b_j^t - g_{ij}^T(A_j^t)^{-1}(w\lambda_0 I_d + (D_j^t)^TD_j^t)(\theta^*-\theta_0)| \\ &= |g_{ij}^T(A_j^t)^{-1}(D_j^t)^T(V_j^t - D_j^t(\theta^*-\theta_0)) - w\lambda_0 g_{ij}^T(A_j^t)^{-1}(\theta^*-\theta_0)| \\ &\leq |g_{ij}^T(A_j^t)^{-1}(D_j^t)^T(V_j^t - D_j^t(\theta^*-\theta_0))| + w\lambda_0 M\left\|(A_j^t)^{-1}g_{ij}\right\|_2, \end{aligned} \tag{26}$$

where the first equality follows from the lazy training process $\hat{c}_{ij} = g_{ij}^T(\theta^t-\theta_0)$ (by Lemma 5), and the second from the identity $A_j^t = w\lambda_0 I_d + (D_j^t)^TD_j^t$ and $b_j^t = (D_j^t)^TV_j^t$. The inequality of (26) follows from the triangular inequality and $\|\theta^*-\theta_0\|_2\leq M$ (by Lemma 5, we take the value of M at $M=\sqrt{2\boldsymbol{c}^T\boldsymbol{H}\boldsymbol{c}/w}$). Because $\mathbb{E}[V_j^t - D_j^t(\theta^*-\theta_0)] = 0$, Azuma–Hoeffding inequality implies the following concentration inequality on the first term of (26).

$$\begin{aligned} \mathbb{P}\left[|g_{ij}^T(A_j^t)^{-1}(D_j^t)^T(V_j^t - D_j^t(\theta^*-\theta_0))| \geq \sqrt{\frac{1}{2}\log\frac{2}{\Delta}}s_{ij}^t\right] &\leq 2\exp\left(-\frac{\log(2/\Delta)(s_{ij}^t)^2}{\left\|D_j^t(A_j^t)^{-1}g_{ij}\right\|_2^2}\right) \\ &\leq 2\exp(-\log(2/\Delta)) = \Delta, \end{aligned} \tag{27}$$

where the second inequality follows from

$$\begin{aligned} \left\|D_j^t(A_j^t)^{-1}g_{ij}\right\|_2^2 &= (D_j^t(A_j^t)^{-1}g_{ij})^TD_j^t(A_j^t)^{-1}g_{ij} \\ &\leq g_{ij}^T(A_j^t)^{-1}(D_j^t)^T(I_d + (D_j^t)^TD_j^t)(A_j^t)^{-1}g_{ij} \\ &= g_{ij}^T(A_j^t)^{-1}g_{ij} = (s_{ij}^t)^2 \end{aligned} \tag{28}$$

Similarly, we have the bound $\left\|(A_j^t)^{-1}g_{ij}\right\|_2 \leq s_{ij}^t$. Combining the above two inequalities (27) and (28), we have, with probability at least $1-\Delta$, $|\hat{c}_{ij}^t - c_{ij}| \leq \left(w\lambda_0 M + \sqrt{\frac{1}{2}\log\frac{2}{\Delta}}\right)s_{ij}^t$. Notice that the gradient satisfies

that $g_{ij} \le \sqrt{wL}$ for all i and j (see Cao and Gu 2019), and the regularization parameter satisfies $\lambda_0 \le O(\sqrt{1/2w\boldsymbol{c}^T\boldsymbol{H}\boldsymbol{c}})$. Therefore, the regularization term satisfies

$$w\lambda_0 M \left\|(A_j^t)^{-1} g_{ij}\right\|_2 \le O(w\sqrt{1/(2w\boldsymbol{c}^T\boldsymbol{H}\boldsymbol{c})} \cdot \sqrt{2\boldsymbol{c}^T\boldsymbol{H}\boldsymbol{c}/w} \cdot s_{ij}^t = O(s_{ij}^t),$$

where the inequality follows from that $\lambda_0 \le O(\sqrt{1/(2w\boldsymbol{c}^T\boldsymbol{H}\boldsymbol{c})})$ and $M = \sqrt{2\boldsymbol{c}^T\boldsymbol{H}\boldsymbol{c}/w}$. Let $\Delta := T^{-4}$. We have, with probability at least $1 - T^{-4}$ and a fixed context i,

$$|\hat{c}_{ij}^t - c_{ij}| \le O(\sqrt{\log T} s_{ij}^t),$$

where $s_{ij}^t = \sqrt{g_{ij}^T (w\lambda_0 I_d + \sum_{i' \in \mathcal{H}_j^t} g_{i'j} g_{i'j}^T)^{-1} g_{ij}}$. By Theorem 2 in Hsu et al. (2014), we integrate over the random set $\mathcal{H}_j^t$ with $n_j^t \ge \Omega(d \log T)$, and obtain that $s_{ij}^t = O(d/n_j^t)$. Therefore, for a fixed context i, with probability at least $1 - T^{-4}$, we have

$$|\hat{c}_{ij}^t - c_{ij}| \le O\left(\sqrt{\frac{d \log T}{n_j^t}}\right) \tag{29}$$

Finally, we take the union bound for all the contexts $1 \le i \le m$ and obtain that: With probability at least $1 - T^{-4}$,

$$|\hat{c}_{ij}^t - c_{ij}| \le O\left(\sqrt{\frac{d \log T + 4d \log m}{n_j^t}}\right) = O\left(\sqrt{\frac{d \log T}{n_j^t}}\right),$$

where the inequality follows from inequality (29) and the union bound, and the equality from that $m \le O(T^{1/4})$. This concludes the proof of Proposition 1. □

G.2. Training with the Gradient Descent Algorithm

We now consider the gradient-based training procedure for neural networks to validate Assumption 2. In fact, one can devise a gradient descent training algorithm that achieves the same convergence rate as lazy training (i.e., $|\hat{c}_{ij}^t - c_{ij}| \le O(\sqrt{d \log T / n_j^t})$), because the training trajectory path, $\{\theta^t\}_{t=1}^t$, of the gradient descent procedure is close to that of lazy training. Formally, we propose the gradient-based training of a neural network as follows.

Training a Neural Network with Gradient Descent

Input: Step size η, number of gradient descent steps U, network width w, regularization parameter λ_0.

Loss function: $\mathcal{L}(\theta) := \sum_{i \in \mathcal{H}} (H_j(x_i, \theta) - v_{ij})^2/2 + w\lambda_0 \|\theta - \theta_0\|_2^2/2$

For $u = 0, 1, 2, \ldots, U-1$ **do**

$\theta^{u+1} = \theta^u - \eta \nabla \mathcal{L}(\theta^u)$

The following proposition shows that Assumption 2 holds for a neural network if trained with the gradient descent algorithm described above.

PROPOSITION 2 **(Prediction Oracle with Gradient-based Training)**. *Under Assumption 3 and all the conditions of Proposition 1, the predicted CTR in each round t, $\hat{c}_{ij}^t$, is obtained by the gradient descent*

algorithm. We have there exist a family of constants $\{C_i\}_{i=0}^5 > 0$ *such that for any* $\delta \in (0,1)$*, if for all* $t \in [T]$*, the regularization parameter* λ_0*, training step size* η*, number of steps* U*, and network width* w *satisfy*

$$
\begin{aligned}
&w \geq C_0 m^4 L^6 \log(m^2 L/\delta)/\gamma^4 \\
&2\sqrt{t/(w\lambda_0)} \geq C_1 w^{-3/2} L^{-3/2} [\log(mL^2/\delta)]^{3/2} \\
&2\sqrt{t/(w\lambda_0)} \leq C_2 \min\{L^{-6}[\log w]^{-3/2}, (w(\lambda_0\eta)^2 L^{-6} t^{-1} (\log w)^{-1})^{3/8}\} \\
&\eta \leq C_3 (w\lambda_0 + twL)^{-1} \\
&U > C_4 \log(d\log(T))/\log(1-\eta w\lambda_0) \\
&w^{1/6} \geq C_5 \sqrt{\log w} L^{7/2} t^{7/6} \lambda_0^{-7/6} (1+\sqrt{t/\lambda_0}),
\end{aligned}
$$

with probability $1-\delta$*, it holds that, if ad* j *is displayed* $n_j^t \geq \Omega(d\log T)$ *times before round* t *and the random click-through outcomes* $\{v_s(a_s) \in \{0,1\} : 1 \leq s \leq t-1\}$ *are observed, then with probability at least* $1-T^{-4}$*, the following inequality holds for any context* $x_i \in X$*:*

$$
|\hat{c}_{ij}^t - c_{ij}| \leq O\left(\sqrt{\frac{d\log T}{n_j^t}}\right).
$$

Before proving Proposition 2, we first introduce Lemma 6 and Lemma 7 below to bound the training trajectory $\{\theta^t : t = 1, 2, ..., T\}$ and the gradient $\nabla_\theta H_j(x_i, \hat{\theta})$, respectively.

LEMMA 6 **(Lemma B.2 in Zhou et al. (2020))**. *There exist a family of constants* $\{C_i\}_{i=1}^5 > 0$ *such that for any* $\delta \in (0,1)$*, if for each* $t \in [T]$*,* η *and* w *satisfy*

$$
\begin{aligned}
&2\sqrt{t/(w\lambda_0)} \geq C_1 w^{-3/2} L^{-3/2} [\log(mL^2/\delta)]^{3/2} \\
&2\sqrt{t/(w\lambda_0)} \leq C_2 \min\{L^{-6}[\log w]^{-3/2}, (w(\lambda_0\eta)^2 L^{-6} t^{-1} (\log w)^{-1})^{3/8}\} \\
&\eta \leq C_3 (w\lambda_0 + twL)^{-1} \\
&w^{1/6} \geq C_4 \sqrt{\log w} L^{7/2} t^{7/6} \lambda_0^{-7/6} (1+\sqrt{t/\lambda_0})
\end{aligned}
$$

then, with probability at least $1-\delta$ *over the He Initialization of* θ_0*, we have, for any* $t \in [T]$*,* $\|\theta^t - \theta_0\|_2 \leq 2\sqrt{t/w\lambda_0}$ *and*

$$
\|\theta^t - (A_j^t)^{-1} b_j^t - \theta_0\|_2 \leq (1-\eta w\lambda_0)^{U/2} \sqrt{t/(w\lambda_0)} + C_5 w^{-2/3} \sqrt{\log w} L^{7/2} t^{5/3} \lambda_0^{-5/3} (1+\sqrt{t/\lambda_0}). \tag{30}
$$

LEMMA 7 **(Lemma B.4 in Zhou et al. (2020))**. *There exist a family of constants* $\{C_i\}_{i=1}^3 > 0$ *such that for any* $\delta \in (0,1)$*, if* τ *satisfies that*

$$
C_1 w^{-3/2} L^{-3/2} [\log(mL^2/\delta)]^{3/2} \leq \tau \leq C_2 L^{-6} [\log m]^{-3/2}.
$$

then, with probability at least $1-\delta$ *over the He Initialization of* θ_0*, for all* $\hat{\theta}$ *and* $\tilde{\theta}$ *satisfying* $\|\hat{\theta} - \theta_0\|_2 \leq \tau$ *and* $\|\tilde{\theta} - \theta_0\|_2 \leq \tau$*, we have, for any context* x_i *and ad* j*,*

$$
\left| H_j(x_i, \tilde{\theta}) - H_j(x_i, \hat{\theta}) - \langle \nabla_\theta H_j(x_i, \hat{\theta}), \tilde{\theta} - \hat{\theta} \rangle \right| \leq C_3 \tau^{4/3} L^3 \sqrt{w \log w}.
$$

With Lemma 6 and Lemma 7, we are now ready to prove Proposition 2.

Proof of Proposition 2.

It suffices to consider the union of the high probability cases in Proposition 1, Lemma 6, and Lemma 7. Let us set $\tau = 2\sqrt{t/w\lambda_0}$ in Lemma 7. At round t, after observing n_j^t i.i.d samples of each ad j, for a fixed context x_i, we have the following inequality:

$$\begin{aligned}
|\hat{c}_{ij}^t - c_{ij}| &= |H_j(x_i,\theta^t) - \langle \nabla_\theta H_j(x_i,\theta_0), \theta^* - \theta_0\rangle| \\
&\le |H_j(x_i,\theta^t) - \langle \nabla_\theta H_j(x_i,\theta_0), (A_j^t)^{-1} b_j^t\rangle| + |\langle \nabla_\theta H_j(x_i,\theta_0), (A_j^t)^{-1} b_j^t\rangle - \langle \nabla_\theta H_j(x_i,\theta_0), \theta^* - \theta_0\rangle| \\
&\le |H_j(x_i,\theta^t) - \langle \nabla_\theta H_j(x_i,\theta_0), (A_j^t)^{-1} b_j^t\rangle| + O(\sqrt{d\log T/n_j^t}) \\
&\le |H_j(x_i,\theta^t) - H_j(x_i,(A_j^t)^{-1}b_j^t + \theta_0) + H_j(x_i,\theta_0)| + C_3\tau^{4/3}L^3\sqrt{w\log w} + O(\sqrt{d\log T/n_j^t}) \\
&\le |\langle \nabla_\theta H_j(x_i,\theta_0), -(A_j^t)^{-1}b_j^t - \theta_0 + \theta^t\rangle| + 2C_3\tau^{4/3}L^3\sqrt{w\log w} + O(\sqrt{d\log T/n_j^t}) \\
&\le (1-\eta w\lambda_0)^{U/2}\sqrt{wL}\sqrt{t/w\lambda_0} + O(t^{2/3}w^{-1/6}\lambda_0^{-2/3}L^3\sqrt{\log w}) + O(\sqrt{d\log T/n_j^t}),
\end{aligned} \tag{31}$$

where the equality follows from Lemma 5. The first inequality of (31) follows from the triangular inequality. The second inequality of (31) follows from Proposition 1. The third and fourth inequalities of (31) follow from Lemma 7, the fact that $\|(A_j^t)^{-1}b_j^t\|_2 \le \tau$ (see Lemma C.4 in Zhou et al. 2020), the triangular inequality, and $H_j(x_i,\theta_0) = 0$ by the *He Initialization* of θ_0. The last inequality of (31) follows from Lemma 6 which bounds $\|\theta^t - (A_j^t)^{-1}b_j^t - \theta_0\|_2$ using inequality (30), and the bound on the gradient $\|\nabla_\theta H_j(x_i,\theta_0)\|_2 \le \sqrt{wL}$ (see Cao and Gu 2019, Zhou et al. 2020). With a sufficiently large neural network width w, the second term of inequality (31), $O(t^{2/3}w^{-1/6}\lambda_0^{-2/3}L^3\sqrt{\log w})$, can be bounded by $O(\sqrt{d\log T/n_j^t})$. The first term of (31), $(1-\eta w\lambda_0)^{U/2}\sqrt{wL}\sqrt{t/w\lambda_0}$ converges to 0 at an exponential rate with respect to the number of training steps U. Because $U > \Omega(\log(d\log(T))/\log(1-\eta w\lambda_0))$, $(1-\eta w\lambda_0)^{U/2}\sqrt{wL}\sqrt{t/w\lambda_0}$ is also bounded by $O(\sqrt{d\log T/n_j^t})$. Therefore, for a given context x_i,

$$|\hat{c}_{ij}^t - c_{ij}| \le O\left(\sqrt{\frac{d\log T}{n_j^t}}\right). \tag{32}$$

Finally, we take the union bound for all the contexts $1 \le i \le m$ and obtain that: With probability at least $1 - T^{-4}$,

$$|\hat{c}_{ij}^t - c_{ij}| \le O\left(\sqrt{\frac{d\log T + 4d\log m}{n_j^t}}\right) = O\left(\sqrt{\frac{d\log T}{n_j^t}}\right),$$

where the inequality follows from inequality (32) and the union bound, and the equality from that $m \le O(T^{1/4})$. This concludes the proof of Proposition 2. □